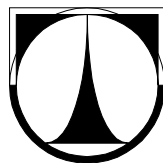


TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií



BAKALÁŘSKÁ PRÁCE

Liberec 2012

Ing. Petr Wolf

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: 1802R022 – Informatika a logistika

**Lokalizace portálu PREDNASKY.TUL.CZ do
českého jazyka**

**Localization portal PREDNASKY.TUL.CZ into
Czech language**

Bakalářská práce

Autor: **Ing. Petr Wolf**

Vedoucí práce: Ing. Martin Vích Vlasák

V Liberci 10. 5. 2012

Tuto
stránku
nahradí
originál

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum: 10.5.2012

Podpis:

Poděkování

Tímto bych chtěl panu Ing. Martinovi Vlasákovi za věnovaný čas a odborné vedení při přípravě a zpracování podkladů, dále pak všem ostatním za cenné rady, které mě dovedli k úspěšnému dokončení bakalářské práce.

Anotace

Řešitel: Petr Wolf

LOKALIZACE PORTÁLU PREDNASKY.TUL.CZ DO ČESKÉHO JAZYKA

Tato bakalářská práce se zabývá možnostmi a následně i vlastním překladem přednáškového portálu Technické univerzity v Liberci. Hlavním smyslem celé práce je zpříjemnit a zjednodušit ovládání daného portálu při řešení nečekaných problémů běžným uživatelům, kterými jsou nejen standardní studenti či učitelé, ale přednáškový portál je také využíván při studiu na univerzitě třetího věku nebo při výuce handicapovaných studentů. Práce je rozdělena do tří částí. První část vysvětluje základní pojmy a současné možnosti vícedruhových médií včetně možného využití. Druhá část se zabývá možnými způsoby lokalizace přednáškového portálu do českého jazyka, vlastním návrhem databáze pro automatický překlad a programem, který prostřednictvím databáze zajišťuje vlastní realizaci. V třetí části je popsána vlastní práce s programem a jeho použití při tvorbě lokalizačních balíčků pro používané moduly.

Počet stran: 49

Počet příloh: 3

Počet obrázků: 44

Počet tabulek: 0

Annotation

Solver: Petr Wolf

LOCALIZATION PORTAL PREDNASKY.TUL.CZ INTO CZECH LANGUAGE

This thesis deals with possibilities and after that actual translation of lecture site of the Technical University in Liberec. The main purpose of this work is sweeten and simplify the operation of the portal during solving of unforeseen problems to common users, which are not only standard students or teachers, but lecture site is used during study of university of third age or by handicapped students. This work is divided on three parts. First explains basic concepts and current opportunities of reach medias including of possible application. Second part deals with ways of localization of lecture site to Czech language, own design of database for automatic translation and by program, which maintains own realization by database. Third parts describes work with program its using during creation of localization packs for using modules.

number of pages: 49

number of supplements: 3

number of figs: 44

number of tablets: 0

OBSAH

Použité zkratky a symboly	8
1. Úvod	10
2. Vícedruhová média	11
2.1 Informace a komunikace	11
2.2 Teorie informace, komunikace	12
2.3 Vícedruhová média	14
2.4 Teorie vicedruhových médií	15
3. MediaSite system	17
3.1 Co je MediaSite systém	17
3.2 Charakteristika MediaSite	18
3.3 Základní parametry MediaSite systému.....	19
3.4 Příklady použití v praxi	20
4. Česká lokalizace systému MediaSite	21
4.1 XML a jeho historie	21
4.2 XML a jeho vlastnosti	22
4.3 Co znamená lokalizace MediaSite systému	23
4.4 Struktura XML v MediaSite	24
5. Návrh programu XML Convert	26
5.1 Rozbor zadání	26
5.2 Návrh databáze	27
5.3 Návrh uživatelského rozhraní	29
5.4 Návrh vyhledávání v databázi	30
5.5 Popis programu XML Convert	36
6. Tvorba lokalizačních balíčků	38
6.1 Lokalizace prostředí	38
6.2 Rozšíření programu o překlad JavaScriptových souborů	46

7. Závěr	47
Seznam literatury	48
Seznam příloh	49

Použité zkratky a symboly

1080i HD	High-definition – formát televizního signálu ve vysokém rozlišení. 1080 je počet řádků.
Access	Program pro správu databází od firmy Microsoft.
ADO	Technologie ADO - ActiveX Data Object, zajišťuje přístup k datovým zdrojům.
AMX™	Jméno výrobce řídicích systémů umožňujících propojovat různá koncová či ovládací zařízení.
Android	Operační systém pro mobilní zařízení.
BDE	Rozhraní BDE – Borland Database Engine umožňuje vytvořit jedno univerzální rozhraní pro různé databáze
BlackBerry	Výrobce mobilních zařízení a softwarů k nim.
CGA	Sdružení CGA (Graphics Communications Association) vyvinula univerzální formátovací jazyk zvaný GenCode.
Crestron®	Jméno výrobce řídicích systémů umožňujících propojovat různá koncová či ovládací zařízení.
CSV	CSV je souborový formát (Comma-separated values) používaný pro operace s tabulkovými daty.
DTD	Definice typu dokumentu (Document Type Definition).
EasyPHP	Program umožňující spouštět programy v PHP na PC, kde simuluje činnost serveru.
Excel	Tabulkový procesor od firmy Microsoft.
FMEA	Je zkratka Failure Mode and Effects Analysis, která znamená analýzu příčin důsledků vad.
NTSC, PAL	Jedná se o standardy standard kódování analogového televizního signálu. NTSC (National Television System(s) Committee) PAL (phase alternating line)
HTML	Značkovací jazyk pro programování webových stránek.
IBM	Významná celosvětová společnost v oblasti informačních technologií.
IEEE 1394	Standard pro vysokorychlostní komunikaci po sériové sběrnici často používaný pro přenos zvuku a videa.
IPad, iPhone, iPod	Jedná se o multifunkční zařízení od firmy Apple (tablet, telefon multimediální přehrávač).
ISO	Mezinárodní organizace pro normalizaci (International Organization for Standardization)
Linux	Unixový operační systém osobních počítačů.
Mac	Osobní počítač od firmy Apple.
Mbd	Je souborový formát databázového programu Access.
MediaSite	MediaSite systém je komplexní zařízení, které umožňuje záznam zvuku, videa a jiných periférií
MSSQL	Databázový program pracující na MSSQL Serverech.

MySQL	Databázový program.
PC	Osobní počítač kompatibilní s PC IBM
QFD	Je zkratka Quality Function Deployment. Tato metoda rozpracovává požadavky zákazníků.
SQL	Structured Query Language – datazovací jazyk pro práci s databázemi.
Tag	Výraz užívaný pro značky ve značkovacích jazycích.
TEX	Jeden z prvních programovacích jazyků určený pro tvorbu prezentací.
UTF-8	Jeden ze způsobů kódování jazykových mutací.
XML	Jedná se o programovací značkový jazyk (eXtensible Markup Language).

1. Úvod

Tato bakalářská práce se zabývá možnostmi a následně i vlastním překladem přednáškového portálu Technické univerzity v Liberci. Hlavním cílem celé práce je zpříjemnit a zjednodušit ovládání portálu prednasky.tul.cz při řešení nečekaných problémů především běžnému uživateli, kterými jsou nejen standardní studenti či učitelé, ale přednáškový portál je také využíván při studiu na univerzitě třetího věku nebo při výuce handicapovaných studentů. Druhým cílem je zjednodušení a urychlení práce správci systému při aktualizaci systému MediaSite. Při každé aktualizaci dochází často ke změně lokalizačních souborů a proto je zapotřebí nástroj, který umožní rychlé a jednoduché generování nových lokalizačních souborů. Druhým důvodem potřeby automatizovaného překladu lokalizačních souborů je jednotnost překladu. To znamená, že při softwarovém řešení překladu, bude při každé aktualizaci překlad identický, stejná hláška stejný překlad. Při ručním překladu se například stávalo, že anglický výraz cancel byl jednou přeložen jako zrušit, jednou jako stornovat nebo jako zavřít.

Práce je rozdělena do tří částí. První část vysvětluje základní pojmy, seznamuje s teorií vícedruhových médií a podává ucelený obraz o možnostech současných vícedruhových médií včetně možného využití v praxi. Druhá část se zabývá možnými způsoby lokalizace přednáškového portálu do českého jazyka, vlastním návrhem a výběrem databáze pro automatický překlad. Dále popisuje strukturu navrženého programu, který prostřednictvím databáze zajišťuje vlastní realizaci automatického překladu do českého jazyka. V třetí části je popsána vlastní práce s programem nazvaným XML Convert a jeho použití při tvorbě lokalizačních balíčků pro používané moduly systému MediaSite na Technické univerzitě v Liberci.

2. Vícedruhová média

2.1 Informace a komunikace

Základním stavebním kamenem všech médií je informace a jejich přenos. Pokud se rozhledneme kolem sebe, všude vidíme nějaké informace, přičemž jejich charakter může být velice odlišný. Informace může být zvuková, vizuální, tepelná, citová apod. Výraz informace z latiny znamená zpráva nebo smysl. Z možných definic informace bych vybral asi tu nejcitovanější od amerického matematika a profesora technologického institutu Massachusetts Norberta Wienera (1894-1964):

„Informace je název pro obsah toho co si vyměníme s vnějším světem, když se mu přizpůsobíme a působíme na něj svým přizpůsobováním.“

[6]

Norbert Wiener se zabýval informacemi a teorií přenosu zpráv už v době druhé světové války. Studoval chování a komunikaci různých systémů (biologickým, nebiologickým, sociálních), snažil se ukázat i jiný pohled na význam slova informace a výsledkem bylo definování obecných pravidel při řízení a komunikování, což se stalo základem pro nový vědní obor – kybernetiku, kde snažil vysvětlit pochopení významu informace pro lidstvo:

"Společnost lze pochopit jedině prostřednictvím studia informace a příslušných sdělovacích prostředků a budoucímu rozvoji těchto informací a sdělovacích prostředků mezi člověkem a strojem, mezi strojem a člověkem a mezi strojem a strojem je souzeno, aby měl stále větší a větší úlohu."

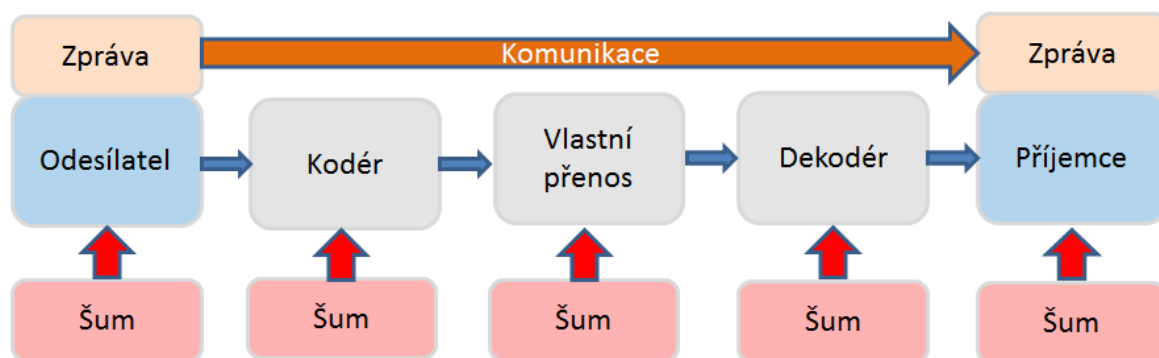
[6]

Informace a jejich vnímání je velice individuální. Například z pohledu přírodních věd můžeme v dnešní době dokonce informaci chápat i jako fyzikální veličinu, kterou můžeme měřit, uchovávat, přenášet nebo přeměňovat. Informaci pak chápeme jako míru snižování nejistoty systému. Naopak z pohledu sociálního je to prostředek myšlení, umožňuje vytvářet představy, zkušenosti, informace používáme k působení na okolí a jsou podkladem při rozhodování. Pohledem filozofa je informace věc jako každá jiná, jeho nezajímá její význam, on touží poznat její podstatu.

Význam informací je v současnosti stále větší a toto slovo se skloňuje ve všech pádech. Jedná se o fenomén tak typický pro současnou dobu, že vznikají úvahy o tom, zda člověk dokáže odolat náporu informací, které na něj chrlí společnost v tištěné, vizuální i audiální podobě nebo prostřednictvím počítačů. [7]

2.2 Teorie informace, komunikace

Významným spolupracovníkem Norberta Wienera byl Claude Shannon (1916-2001), který je dnes považován za zakladatele teorie informace. Jeho nejvýznamnější dílo Matematická teorie komunikace (A mathematical theory of communication) bylo tvořeno během druhé světové války, kdy bylo třeba určit, jaká je pravděpodobnost, že bojová letadla létající do nepřátelské palby za atmosférických poruch zachytí vysílané rozkazy a zprávy v té podobě, v jaké byly vyslány. Vyšel z myšlenky, že máme-li dvě možnosti a dozvíme se, že jedna z nich platí, získáme nejmenší potřebné množství informací - vytvořil tak jednotku informace BIT (binary digit = dvojkové číslo) - elementární rozhodnutí mezi ano a ne. Shannon matematicky vyjádřil množství hlásek vysílané zprávy, které se při přenosu zprávy ztratí a zprávu deformuje - šumy. Mezi nejznámější objevené zákonitosti této teorie patří: informační šumy, jednotka informace, vymezení prostředků informace, nutnost zakódování informace, redundance informace. Mezi další poznatky Shannonovy teorie patří i definování prostředků nutných pro přenos informací: vysílač, přijímač, kanál. Chápe, že zprávu je třeba zakódovat do určitého jazyka (ať již přirozeného nebo umělého) - kódu. [7]



Obrázek 2.1: Komunikační řetězec

Vytvořil tak základní technické poznatky přenosu informace. Na obrázku 2.1 je zobrazeno schéma komunikačního řetězce. Smyslem je zajistit přenos požadované informace od odesílatele k příjemci. Odesílatel odešle svou zprávu, to znamená, že vysílač zakóduje zprávu a odešle ji přes přenosové médium. Problém ale nastává v případě rušení signálu (šumu), který „deformuje“ odesílanou zprávu. Šum může „napadnout“ přenášenou zprávu kdykoliv během přenosu. Pod pojmem šum si nejčastěji představíme chyby při přenosu zprávy, ale také to může být chyba odesílatele, který špatně zprávu zapíše. Na druhé straně dekodér musí obnovit příchozí zprávu do původního stavu (musí eliminovat šum), a tím získá příjemce nezkreslenou odesílanou zprávu.

Pod pojmem komunikace můžeme najít opět různé definice. Výraz komunikace je původem z latinského slova *cumunicare*, což znamená sdílet či radit se. Z pohledu vícedruhových médií je potřebné si uvědomit, že komunikace není jen verbální a je i neverbální. Jako jeden z prvních si uvědomil už Norbert Wiener.

Sdělování mezi lidmi se podle Wienera odlišuje od sdělování u většiny ostatních živočichů jemností a složitostí používaného kódu a značně vysokou konvenčností tohoto kódu (slovní - verbální komunikace). Pro příjem sémantické informace je zapotřebí paměti. Jde tedy o proces abstrakce a fonetizace, při níž se vytvářejí synapse v našem mozku. Do pochopení obsahu zprávy každý člověk vkládá svou minulou zkušenost. [Slámová, 2010]

Tyto teorie později rozpracoval americký psycholog a filozof rakouského původu Paul Watzlawick (1921-2007). Watzlawick (1999) rozlišil komunikaci digitální a analogovou. „Digitalni komunikace“ se opírá o rozum a logiku a lze ji bez ztráty informace zapsat. Obvykle se jedná o komunikaci verbální. „Analogova komunikace“ je obrazná a intuitivní. Často jde o sdělování prostřednictvím postoje, činem. Informace je předávána osobním příkladem. Lze ji jen těžko bezezbytku převést do psané podoby. [8]

Neverbální komunikace určuje, jak má být verbální sdělení chápáno. Jedna starší studie ilustruje význam mimoslovních sdělení. Americký emeritní profesor psychologie Albert Mehrabian (1972) uskutečnil výzkum, kterým zjišťoval, podle čeho lidé usuzují o druhých lidech. Zjistil, že verbální projev ovlivní 7 % celkového dojmu. Modulace hlasu pak 38 % a mimika pak zbylých 55 %. Zjistil také, že pokud jsou verbální a neverbální projevy v rozporu, je větší váha přisouzena projevům neverbálním. Vedle toho, že neverbální chování je svébytným komunikačním kanálem, plní často funkci doprovodnou. Doplnuje, pozměňuje a rozděluje verbální sdělení. [8], [9]

Toto zjištění má velký význam v souvislosti s vícedruhovými médii, která nabízí záznam nejen té verbální stránky, ale i těch ostatních. Záznam a následné zobrazení všech mimoverbálních aktivit během přednášky výrazně zvyšují vnímání posluchače. Vždyť cílem každého přednášejícího je, aby si posluchač odnesl z jeho přednášky co nejvíce informací. Na výše uvedený výzkum se můžeme podívat i z druhé strany. To znamená, že lidé více vyhledávají přímý kontakt s osobou, se kterou komunikují, aby mohli lépe sledovat jejich reakci, jejich mimiku. Tento způsob komunikace se nazývá Face to Face. Dle průzkumu, který provedla společnost HR Solution [10] vyplývá, že více jak 50% lidí požaduje přímou komunikaci se svým nadřízeným nebo spolupracovníky a jen 35% dává přednost e-mailům a důvodů je několik. Tím hlavním důvodem jsou přímé informace s možností okamžité reakce a lidé jsou si k sobě o něco blíže z pohledu lidského a osobního. Tento způsob komunikace není možné v současnosti plně rozvinout ve vícedruhových médiích, neboť se jedná pouze o záznam, ale na něm je uživatel v přímém kontaktu s přednášejícím a tím má alespoň pocit komunikace Face to Face.

2.3 Vícedruhová média

Nejprve se podíváme na to, co znamená pojem vícedruhová média neboli původní anglický výraz Rich media. Slovo media znamená, podle slovníku českého jazyka množné číslo od slova médium, jehož význam je zprostředkující, výplň prostoru nebo látka, ve které je předmět uložen. Běžně užívaný výraz multimédia pak znamená širokou škálu digitálních interaktivních (neboli vzájemně komunikujících) médií. Vícedruhová (Rich) média navíc umožňují stažení záznamu nebo možnost vložení do webové stránky. Vlastní stažení může být provedeno on-line nebo off-line a vlastní prohlížení pak probíhá na běžných prohlížečích jako např. RealPlayer, Microsoft Media Player nebo Apple QuickTime.

Charakteristickým znakem vícedruhových médií je to, že vykazují dynamický pohyb. K tomuto pohybu může dojít v průběhu času nebo v přímé odezvě na interakci uživatele. Uvedu dva příklady dynamického pohybu, které se vyskytují v průběhu času - akce streamování videa a akce "ticker", která umožňuje průběžně vlastní aktualizaci. Příkladem dynamického pohybu v odezvě na interakci uživatele je možnost řízení on-line přenosu předem nahraných záznamů spolu se synchronizovanou prezentací. Dalším příkladem je animovaný, interaktivní soubor (prezentace) vložená do webové stránky.

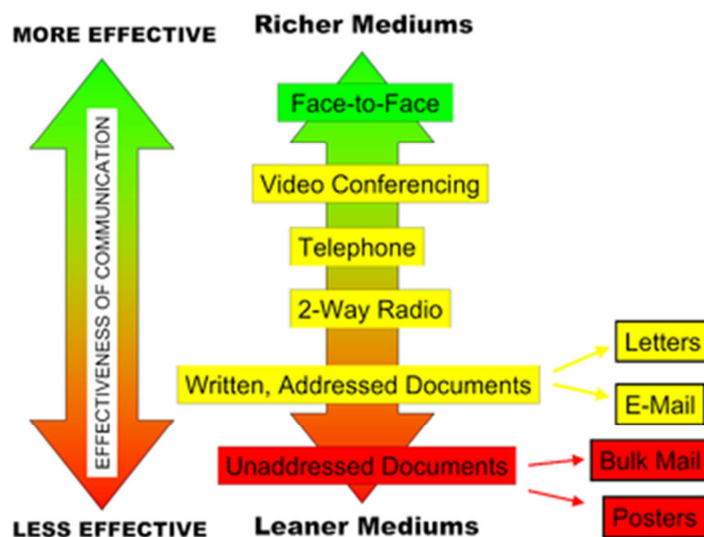
Prvky vícedruhových médií jsou stále více využívány v oblasti vzdělávání, v oblasti dálkového studia, webové výuky nebo i učebních pomůcek. Není divu, že vícedruhová média přináší řadu návrhů pro snadný přístup. Nicméně multimédia jsou zpřístupněna, pokud všechny prvky jsou vyvinuty s ohledem na přístupnost a pak i konečný produkt je používán nebo prohlížen na dostupných multimediálních přehrávačích. Přístupná vícedruhová média obvykle zahrnují titulky, zvukový popis a navigaci pomocí klávesnice. Přístupné multimediální přehrávače jsou takové, které lze ovládat všemi uživateli, včetně těch, které je využívají jen pro čtení záznamů. Musí také poskytnout autorům prostředky, jak přidat titulky, zvukové popisy nebo audio rozšířené popisy a titulky. Současná úroveň přístupnosti přehrávačů médií vytváří zajímavé situace. Některé multimediální přehrávače umožňují přidat videa, ale rozhraní jim neumožňuje přidat popisy. To má za následek chybu při přehrávání těchto popisů. Dalším problémem je to, že popisky mohou vypadat jinak, když jsou vytvořeny na jednom přehrávači a následně se přehrávají na jiném. Například, titulky vyvinuté pomocí QuickTime budou vypadat dobře při zobrazení v QuickTime, ale v aplikaci RealPlayer se pak objeví větší nebo menší. I v této oblasti dochází k výraznému zlepšení, kdy prostřednictvím programu MAGpie máme k dispozici snadný nástroj pro přidání titulků a zvuku při popisu video a audio souborů. Stranou nezůstaly ani programátoři jednotlivých přehrávačů a i oni se snaží zlepšit a přizpůsobit své produkty tak, aby lépe vyhovovaly dnešním požadavkům z pohledu vícedruhových médií. [20]

2.4 Torie vícedruhových médií

Teorie vícedruhových médií jsou důsledkem rozvoje komunikačních technologií. Je potřeba si uvědomit, že každý způsob komunikace je trochu jiný, jinak působí na jedince a jiné má důsledky na vlastní komunikaci. Aby bylo možné lépe pochopit tento problém, je potřeba porozumět tomu, jak působí jednotlivé komunikační technologie a jaké mají důsledky, aby bylo možné následně navrhnout optimální komunikační technologii pro danou skupinu uživatelů. Mezi nejvýznamnější teorie zabývající se komunikací a médii patří Media Richness Theory a Media Naturalness Theory.

Media Richness Theory neboli teorie vícedruhových médií. Tuto teorii navrhoval Richard Daft a Robert Lengel v roce 1984, v podstatě říká, že výsledek daného úkolu se zvyšuje, pokud jsou úkolu přizpůsobeny možnosti vícedruhových médií. Jinými slovy, pokud můžu použít médium s vyšší úrovní "bohatství", které se nejlépe hodí pro daný úkol, pak informace o úkolu pro jednotlivce bude vyšší a výkon jedince tohoto úkolu bude také vyšší. Navíc, teorie naznačuje, že jednotlivci očekávají použití specifických komunikačních prostředků k plnění určitých úkolů. Je tím myšleno, že vícedruhová média jsou schopná s větší pravděpodobností nalézt vhodnou cestu k vyjasnění "nejasné" komunikace, která se vyskytuje většinou při řešení složitých úkolů.

Daft a Lengel definovali "bohatství" médií jako schopnost porozumění informacím v určitém časovém intervalu. Teorie tvrdí, že "bohatství" médií se liší mezi typy médií, to znamená, že komunikace typu Face to Face je bohatší než komunikaci prostřednictvím e-mailu. Na obrázku 2.2 je vidět vztah mezi efektivitou porozumění a typem média.

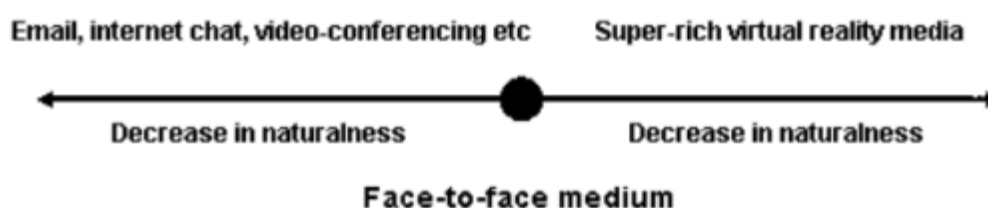


Obrázek 2.2: Media Richness Theory [14]

Slabinou této teorie je, že se snaží porovnávat navzájem jednotlivé typy medií a určit, které z nich je lepší (které umí přenést větší množství informací bez ohledu na jejich obsah). Ale ve skutečnosti má každý druh média své výhody a nevýhody. Tato teorie byla vytvořena ještě v době, kdy nebyl tak široký rozmach internetu, e-mailu nebo chatu, videokonference nebo sociální sítě v podstatě neexistovaly a současné pokusy, které se snaží setřídít média dle této teorie, jsou proto neúspěšné. Kritikou této teorie je i to, že nezahrnuje vliv sociálního

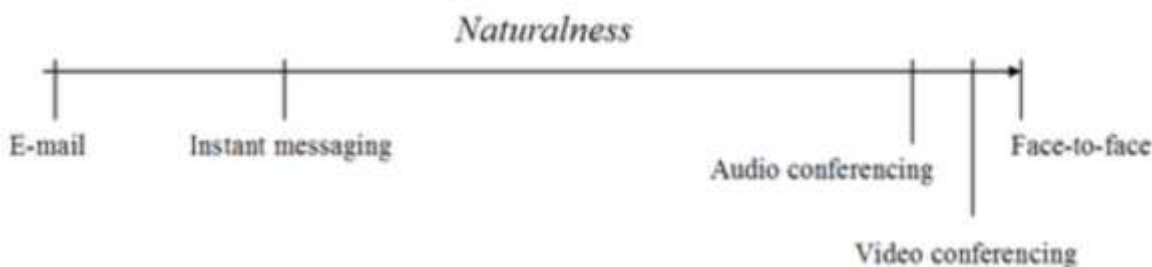
(marketingového) tlaku na média, který je může ovlivňovat více než vlastní “bohatství” daného média. [12] [13] [14]

Media Naturalness Theory neboli teorie o přirozenosti médií. Tato teorie byla navržena jako alternativa k Media Richness Theory. Vychází z předpokladu přirozeného vývoje, kdy lidé už v době kamenné komunikovali pouze Face to Face, a proto se náš mozek vyvíjel a přizpůsoboval tomuto druhu komunikace. Nové formy komunikace jsou zde příliš krátkou dobu na to, aby se jim mohl náš mozek přizpůsobit. Komunikační prostředky, které potlačují klíčové prvky obsažené v komunikaci Face to Face, představují tak kognitivní překážky v komunikaci. Toto pak zejména platí při řešení složitých a komplexních úkolů.



Obrázek. 2.3 Přirozenost médií vzhledem k Face to Face [15]

Media Naturalness Theory používá komunikace typu Face to Face jako základního typu komunikace a všechny ostatní druhy jsou s ní srovnávány. Pokud definuje přirozenost média jako stupeň podobnosti k Face to Face pak můžeme vytvořit jednorozměrnou stupnici, kde měřítkem je přirozenost média. Toto je dobře vidět na obrázku 2.3. Pokud se nyní pokusím srovnat média dle přirozené schopnosti předávání informací, pak je zřejmé, že nejvýznamnějším typem komunikace bude opět Face to Face, pak budou následovat video a audio konference, protože člověk je zvyklý sledovat a poslouchat svého komunikačního partnera a nejmenší přirozenou schopnost předávat informace bude mít e-mailová komunikace. Obrázek 2.4 představuje přirozenost médií v měřítku.[15]



Obrázek. 2.4 Přirozenost médií v měřítku [15]

3. MediaSite systém

3.1 Co je MediaSite systém

MediaSite systém je komplexní zařízení, které umožňuje záznam zvuku, videa a jiných periférií jako například prezentací v reálném čase. To znamená, že prováděný záznam je možné zaznamenat ve vysokém rozlišení nebo on-line přenášet přes webové rozhraní. Součástí je i správa a přehrávání vytvořených záznamů prostřednictvím webových stránek. Tento způsob rozhraní umožňuje rychlé a masové rozšíření jednotlivých záznamů k cílovým uživatelům. Každý záznam je možné svázat s oprávněním umožňující jeho zpřístupnění v internetové síti. Z obrázku 3.2 je možné si udělat představu, jak taková webová aplikace na prohlížení záznamu může vypadat.



Obrázek 3.1 Logo Mediasite [1]



Obrázek 3.2 Příklad zobrazeného záznamu [1]

Z pohledu struktury můžeme MediaSite systém rozdělit do základních dvou částí – hardware-ové a software-ové .

1. Hardware-ová část představuje zařízení, ke kterému jsou připojena jednotlivá zařízení jako například kamera, počítač, kamera na snímání psaného textu nebo interaktivní tabule. Toto zařízení je buď instalováno na pevno a je součástí učebny či konferenčního sálu nebo existuje i přenosná verze, která se využívá dle potřeby. Příklad pevného typu (označení RL Recorder) nebo přenosné verze (označení ML Recorder) je na obrázku 3.3a resp. 3.3b.



Obrázek 3.3a RL Recorder [1]



Obrázek 3.3b ML Recorder [1]

Jednou z předností těchto zařízení je možnost připojení již stávajícího zařízení, které uživatel již využívá. RL Recorderu je navíc vybaven sériovým rozhraním RS232, které umožňuje propojení MediaSite systému s řídicími systémy nebo ovládacími panely například od firem AMX, Crestron, AVIT.

2. Software-ová část představuje vlastní realizaci propojení, řízení a ovládání jednotlivých periférií. Konkrétní aplikace se odvíjí od připojených zařízení a objednaných modulů. Cílem aplikace vývojářů bylo vytvořit uživatelsky co nejjednodušší rozhraní pro přednášejícího a následně i pro cílové uživatele, kteří s daným záznamem pracují.

3.2 Charakteristika MediaSite

MediaSite systém v současné době zřejmě pomyslnou jedničkou na trhu ve tvorbě multimediálních záznamů. Ve spojení s MediaSite EX Serverem, provides everything you need to record, deliver, watch and manage your organization's knowledge and special events. poskytuje vše, co je potřeba k nahrávání, sledování a řízení záznamů. Z tohoto uživatelského pohledu můžeme rozdělit MediaSite systém do 4 oblastí.



Obr. 3.4a Modul Record [1]

Modul Record zajišťuje automatické nahrávání, propojení a synchronizaci během nahrávky s PC, notebooky, tablety, interaktivními tabulemi apod. Umožňuje vytvoření plánu nahrávek a jejich automatické spouštění a zastavování a to ve stacionární nebo přenosné verzi.

Deliver



Jedná se o modul umožňující přístup k záznamům kdykoliv a kdekoliv. Modul umožňuje sledovat záznamy na libovolných zařízeních, stáhnout záznamy v různých formátech (MP3, MP4) případně export na externí média (CD, DVD, USB) opět s možností vytvoření plánu exportů a tím jejich automatické provádění. Součástí je i export záznamů neboli zobrazení záznamů na libovolných webových stránkách.

Obr. 3.4b Modul Deliver [1]

Watch



Tento modul zajišťuje vlastní přehrávání záznamů. MediaSite přehrávač poskytuje všem uživatelům (studentům, zaměstnancům) jednoduchý přístup k přednáškám nebo prezentacím. Uživatel může sledovat záznamy on-line nebo off-line. Hlavními znaky modulu jsou možnost prohlížení záznamů ve vysokém rozlišení videa a prezentací a interaktivní ovládání či vyhledávání záznamů.

Obr. 3.4c Modul Watch [1]

Manage



Plné využití tohoto modulu je ve spojení s MediaSite EX Serverem a poskytuje uživatelům úplnou správu jejich záznamů. Záznamy jsou strukturovány do knihoven a tím i zajištěn „pořádek“ v uložených datech. Hlavní funkce modulu Manage jsou plánovač, katalog, vyhledávání či editor. Zajímavou funkcí je analýza, která umožňuje uživateli sledovat statistické ukazatele využití zařízení, sledovanost záznamů apod.

Obr. 3.4d Modul Manage[1]

3.3 Základní parametry MediaSite systému

- Procesor na bázi Intel
- Maximální rozlišení 1920 x 1080
- Podpora širokoúhlého rozlišení• 480p, 720p, 1080i HD
- Formáty NTSC a PAL
- Digitální: IEEE 1394 DV
- 160 GB hard disk (uloží až 1000 hodin obsahu), DVD mechanika
- Ethernetový interface (10/100/1000)
- MediaSite EX Server vytváří z MediaSite Recorderu robustní a spolehlivé mediální prostředí pro záznam, stream, archivaci a organizaci vašich znalostí a důležitých událostí.
- On-line přenos přednášky synchronně s prezentací přednášejícího
- Podpora libovolného zařízení - PC, Mac, iPad, iPhone, iPod, BlackBerry, aplikace pro Android nebo Linux

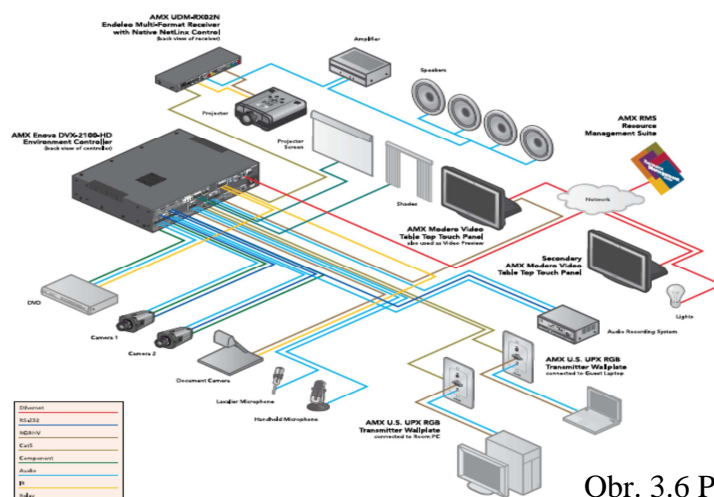
3.4 Příklady využití v praxi

Z výše uvedeného textu je zřejmé, že využití v praxi je velmi široké, tedy všude kde dochází k přednáškám, prezentacím, představení výrobků, nových postupů apod. Jinými slovy jedná se o oblasti vzdělávání, kde se zaznamenávají především přednášky, zdravotnictví kde se prezentují moderní metody a postupy, v průmyslu při představení nových technologií nebo při prezentování vizí a cílů politickými stranami či jednotlivými ministerstvy.



Obrázek 3.5 MediaSite v praxi [21]

V praxi je i významné propojení systému MediaSite s řídicími systémy od firem např. AMX nebo Crestron. Tyto firmy nabízejí velký sortiment řídicích systémů, které propojují jednotlivé periférie včetně ovládacích panelů a rozhraní následně MediaSite systém se stará o nahrávání a správu záznamů. Příklad systému AMX je na obrázku 3.6.



Obr. 3.6 Příklad systému AMX [21]

4. Česká lokalizace systému MediaSite

4.1 XMLa jeho historie

XML je zkratka anglických slov eXtensible Markup Language, v češtině to znamená Rozšiřitelný značkovací jazyk. Značkovací jazyk umožňuje označovat části textu pomocí značek a přidělovat ji určité vlastnosti.

Tento druh jazyka se počal vyvíjet již v 60-tých letech minulého století. Jedním z prvních a nejrozšířenějších byl jazyk T_EX, který se využíval především při tvorbě prezentací. Uživatel měl možnost formátovat jednotlivé části textu a tím vytvářet výslednou podobu dokumentu. Níže je uveden příklad, jak je možné v tomto jazyce změnit například písmo na tučné.

`{\bf zde bude vypsán text tučným písmem}`

Tento jazyk se v určité podobě dochoval do dnes. Je určen pro zpracování textových souborů, které se mají následně tisknout. Využívá se především pro přípravu knih, jednotlivé vlastnosti textu je pak možné měnit v závislosti na tom, zda bude kniha tištěná nebo bude vydána v elektronické podobě.

Asi v 80-tých letech minulého století vyvinula firma IBM vlastní značkovací jazyk pro správu textů. Vývojáři jazyk nazvali GML, což je zkratka slov Generalized Markup Language, v českém překladu to znamená Obecný značkovací jazyk. Cílem bylo vytvořit obecný univerzální a především jednoduše čitelný formát jazyka. Problémem této doby byla nekompatibilita mezi jednotlivými systémy, každý si vytvářel své struktury, formáty či znakové sady. Např. sdružení CGA (Graphics Communications Association) vyvinula univerzální formátovací jazyk zvaný GenCode, ale jak se později ukázalo, jeho obecnost v praxi byla nedostatečná. Aby bylo možné větší rozšíření jazyka GML, bylo potřeba vytvořit standardy v oblasti sad značek a formátů, které by byly přenositelné mezi různými programy resp. platformami. Proto americká standardizační organizace ANSI (American National Standards Institute) ve spolupráci s CGA v rámci projektu ODA (Open Document Architecture) vytvořily jazyk SGML, který byl v roce 1986 schválen ISO (International Standards Organization) v normě ISO 8879.

SGML je předně zcela otevřeným standardem nezávislým na platformách, výrobcích nebo aplikacích. Soubory SGML jsou ukládány jako text ASCII, což zajišťuje jejich použitelnost prakticky na libovolné počítačové platformě.

Hlavní přínos SGML je vytvoření pevné struktury a definování vnitřních vztahů jednotlivých elementů pomocí značek. Uživatel má možnost si tvořit své značky a přiřadit jim libovolný význam. Toto nabízí široké využití při zpracování textu. Představme si příklad, kdy v nějakém textu máme několik odstavců, ale chceme, aby přístup pro čtení různých odstavců byl umožněn i různým uživatelům. V případě značkovacího jazyka pouze stačí příslušné

odstavce označit značkami, které budou charakterizovat příslušnou skupinu uživatelů, kteří mají oprávnění pro čtení. Pak na základě takto umístěných značek v dokumentu se zobrazí každému uživateli již jemu určený text.

Dalším významným prvkem jazyka SGML je definice typu dokumentu, zkráceně DTD (Document Type Definition). Toto umožňuje uživateli definovat typ dokumentu, který je právě zpracováván.

HTML (Hypertext Markup Language) je dnes nejrozšířenější a nejznámější aplikací jazyka SGML pro tvorbu webových stránek. V hlavičce se uvádí DTD definující např. verzi HTML a vlastní program je dán strukturou příkazů resp. značek. Jazyk HTML definuje předem definované skupiny značek (často se využívá anglického výrazu tag) a není možné je rozšířit o své vlastní značky, tak jak to bylo původně zamýšleno při návrhu jazyka SGML. Na příkladu HTML je vidět, že jazyk SGML je příliš obecný a v praxi neuchopitelný. Zahrnout všechny vlastnosti SGML do konkrétního jazyka by bylo příliš složité a v důsledku pro uživatele i velmi náročné a proto se dnes používá pouze určitá část vlastností SGML. Důsledkem tohoto přirozeného vývoje jazyka SGML je vznik jazyka XML (eXtensible Markup Language), který přejímá využívané části jazyka SGML a navíc připojuje nové vlastnosti jako například podpora národních jazyků.

4.2 XMLa jeho vlastnosti

Jazyk XML byl vytvořen pracovní skupinou W3C (World Wide Web Consortium), kteří si stanovili 10 cílů k dosažení úspěchu formátu XML.

1. XML musí být užívaný na internetu.
2. XML musí být využíván i v aplikacích mimo internet.
3. XML musí být kompatibilní s jazykem SGML.
4. Programy (procesy) zpracovávající XML musí být jednoduché.
5. Minimální počet povinných částí by měl být nulový.
6. XML dokument musí být přehledný a snadno čitelný.
7. Vysoká rychlost návrhu XML dokumentu.
8. Návrh XML dokumentu musí být stručný a názorný.
9. Vytváření programů v XML musí být jednoduché.
10. Obsažnost značkování není důležitá.

Definice, cíle, struktura, popis formátu a další je možné nalézt na web. stránkách

<http://www.w3.org/TR/2008/PER-xml-20080205/> .

Jak moc jsou výše uvedené cíle naplňované v praxi, se teprve ukáže časem v závislosti na tom, jak bude tento formát nadále programátory využíván. Většina cílů je zřejmých a

nemá smysl se jimi dále zabírat, ale rád bych se pozastavil u některých z nich, které mají vztah k systému MediaSite.

Cíl 2. V současnosti je formát XML využíván spíše v prostředí internetu pro přenos dat a jako podpora jazyka HTML. Do budoucna se dá očekávat, že na internetu se budou stále více objevovat stránky psané v jazyce XML nebo v XHTML (eXtensible Hypertext Markup Language), jazyk XHTML rozšiřuje jazyk HTML o syntaxi XML. Asi hlavní výhodou jazyka XML je jeho jednoduchá struktura, kterou je možné datově implementovat formou stromu. Dnes asi nejznámější datovou stromovou strukturou pracující s dokumenty je formát DOM (Document Object Model). Pokud datovou strukturu XML implementujeme do stromové struktury DOM, pak můžeme jednoduše dokumentem procházet nebo v něm vyhledávat. Výhodou tohoto způsobu je jednoduché zpracování XML dokumentů a nevýhodou je paměťová náročnost u rozsáhlých datových struktur XML. Míra využití a implementování datové struktury XML bude ovlivňovat naplnění cíle 4. Kromě DOM struktury byly vytvořeny i různé parsery, které mají za úkol podporovat a rozšířit implementování XML formátu do ostatních aplikací.

Program Medisite využívá XML jako podporu pro strukturování názvů zobrazovaných výrazů. Program, který je součástí této práce, má za úkol rozpoznat jednotlivé položky ve struktuře souboru XML a následně je bude zpracovávat.

4.3 Co znamená lokalizace MediaSite systému

Výraz lokalizace znamená určení místa nebo lokality v prostoru resp. v prostředí. V případě, že mluvíme o lokalizaci v souvislosti s nějakým softwarem, pak to znamená přizpůsobení jazyka daného softwaru lokalitě, ve které se daným jazykem mluví. V tomto případě se jedná o přizpůsobení softwaru MediaSite do českého jazyka.

MediaSite standardně využívá anglického jazyka, ve kterém jsou psané všechny nadpisy, hlášky, popisy apod. Tyto výrazy jsou uloženy do souborů s příponou XML. Soubory XML mají jasně definovanou strukturu (více v kap. 4.4 a obrázek 4.1). Popis zprávy je obsažen v tagu s názvem ITEM. Parametr NAME definuje typ zprávy a mezi tagy je zapsán výraz, který se vypisuje při volání daného typu zprávy. Aby to bylo zřejmé, bude to vysvětleno na následujícím příkladu.

```
<item name="UsernameText">Username</item>
```

Kde výraz "Username" bude vypsán v místě programu, kde je požadován typ zprávy "UsernameText". Obdobně vypadá i definice hlášky pro heslo.

```
<item name="PasswordText">Password</item>
```

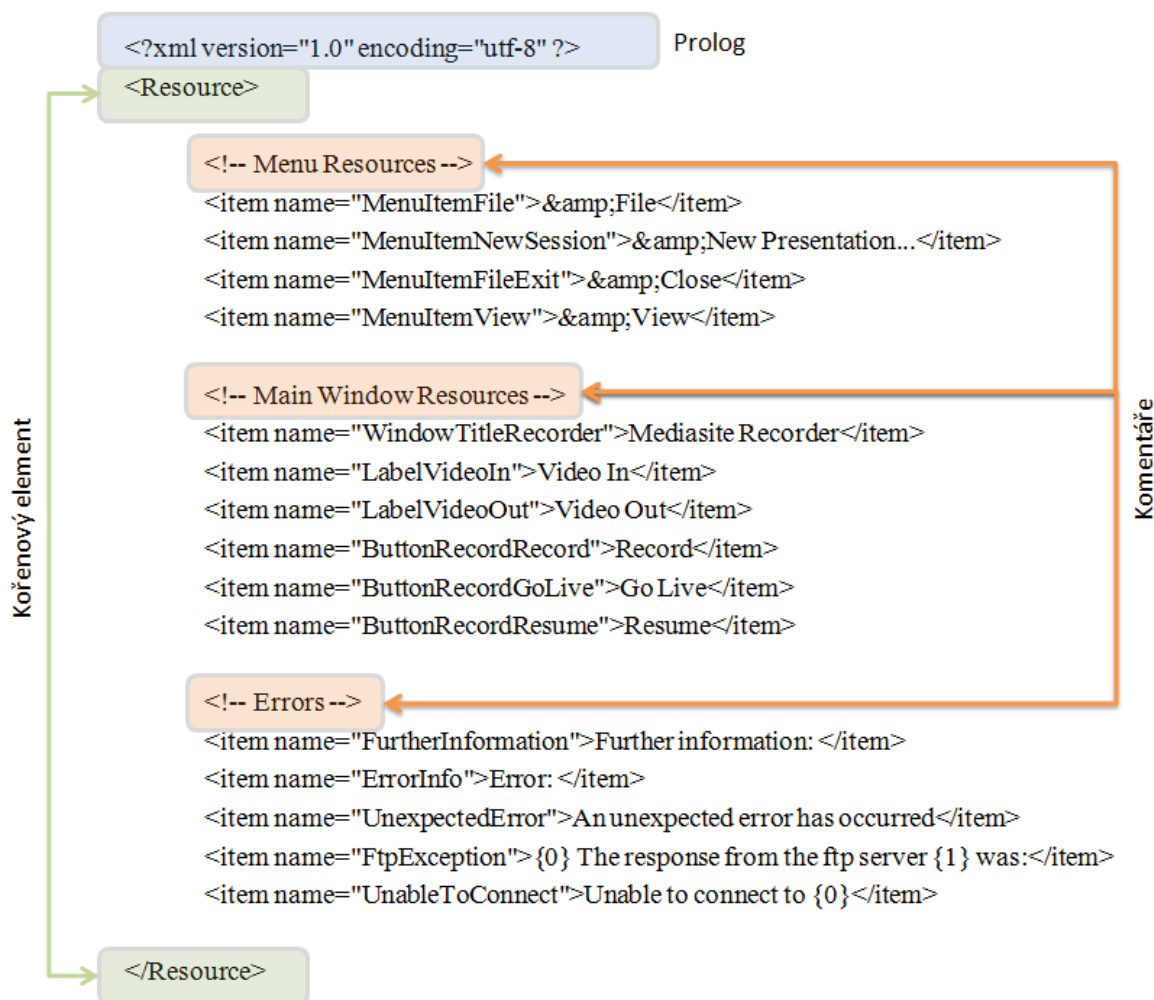

Mezi tagy je možné kromě textu předávat také proměnné programu. Proměnné jsou uvedeny ve složených závorkách a označeny pouze číslem. Konkrétní číslo pak odpovídá konkrétní proměnné. Opět se můžeme podívat na příklad.

```
<item name="Configuration_MediaSiteSectionNotFound">
The configuration file {0} does not contain the section group {1}.</item>
```

V XML struktuře musí na rozdíl od HTML vždy existovat počáteční a koncový tag a takovéto tagy nazýváme párové. V HTML jazyce je možné používat i nepárové tagy, které provádí konkrétní příkaz. Například odřádkování se provádí pomocí tagu
.

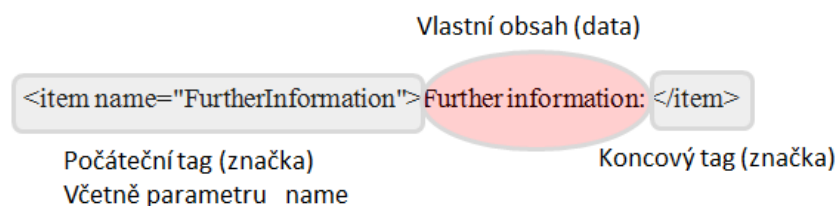
4.4 Struktura XML v MediaSite

Jak již bylo popsáno XML formát je strukturovaný a má jasná pravidla. Některé značky neboli tagy mají konkrétní význam, jiné tvoří strukturu. Následujícím obrázkem 4.1 je výpis konkrétního souboru XML z MediaSite na kterém bude tato struktura vysvětlena.



Obrázek 4.1. Příklad struktury XML

Na výpisu je dobře vidět struktura formátu XML. V úvodu je hlavička někdy nazývána jako prolog a následuje vlastní tělo, kde vlastní obsah je uvozen počáteční a koncový znakem (tagem). V takovém případě mluvíme o párovém tagu. Zde je uveden tag RESOURCE nebo ITEM. V případě, že koncový tag není požadován, pak se jedná o nepárový tag. V tomto případě se jedná o komentář. Každý tag je umístěn do klíčových znaků < a >. Uvozující tag obsahuje vždy název a ukončující tag má vždy před názvem lomítko. Graficky je to zobrazeno na obrázku 4.2.



Obrázek 4.2 Popis elementu

Prolog představuje deklaraci formátu XML, jeho verzi a způsob kódování znaků. V tomto případě je verze specifikace XML 1.0 a jednotlivé znaky jsou kódovány dle znakové sady UTF-8. Tento řádek definující formát XML je sice nepovinný, ale z hlediska přehlednosti se doporučuje ho používat.

Tak jak existuje prolog na počátku souboru, je možné umisťovat položky (elementy) i za kořenový element. Do této oblasti se standardně umisťují jen komentáře nebo instrukce či parametry pro zpracování. Tyto informace pak využívá programem, který následně soubor XML zpracovává.

K vytvoření české lokalizace už zbývá jen načíst soubory XML, rozpoznat tagy s označením ITEM a u nich přeložit výraz v anglickém jazyce, který je umístěn mezi tagy. Za tímto účelem bude vytvořen program v C++, který zajistí rozhraní s uživatelem a vlastní překlad XML souborů. Vlastní překlad bude prováděn pomocí databáze, ve které budou uvedeny názvy typu zprávy (tedy výraz u parametru NAME), anglický výraz dané zprávy a k ní odpovídající ekvivalent v českém jazyce. Aby bylo možné širší využití této databáze i pro případy, kdy nebudou jednotlivé výrazy shodné, bude program mimo jiné porovnávat podobnost hledaných výrazů s výrazy definovanými v databázi a nabízet uživateli (dle předem definovaných pravidel) vhodné návrhy překládaného výrazu. Dále je uvedeno v rozboru zadání programu v kap. č.5.

5. Návrh programu XML Convert

5.1 Rozbor zadání

Součástí práce je návrh programu, který načte obsah příslušného souboru XML a porovná jeho obsah s předem definovanými překlady. Výsledkem bude automatický překlad, nebo program nabídne uživateli výběr z co nejpodobnějších výrazů. Uživatel musí mít případně možnost tento návrh změnit (přepsat) a následně uložit zpět např. do databáze a to jako změněný nebo nový záznam. Po ukončení překladu vypíše program výslednou zprávu o průběhu překladu.

Na základě zadání jsem rozdělil úlohu na několik menších částí, které byly řešeny samostatně, aby se následně spojily do požadovaného výsledku. Rozdělení úlohy umožňuje jednoznačněji si uvědomit nutné kroky v každé části pro naplnění zadání, případně objevit slabá místa návrhu. U velkých projektů, které jsou tvořeny týmy lidí, se doporučuje provádění analýz odhalujících potenciální rizika nedostatku návrhu nebo chyb při zpracování. Mezi nejznámější analýzy patří QFD nebo FMEA. QFD je zkratka Quality Function Deployment, často překládané jako Rozpracování požadavků zákazníka nebo jako Dům kvality. Obecně tato metoda porovnává požadavky zákazníků na výrobek (v tomto případě by se jednalo o porovnání požadavků zadání s jeho návrhem na zpracování). Výsledkem je tabulka, kde jednotlivé požadavky se píší do řádků a do sloupců se zaznamenávají charakteristiky navrhovaného výrobku, které by měly být pokud možno měřitelné, aby je bylo možné následně porovnávat. Vzájemným ohodnocením požadavků a charakteristik je možné určit slabé a silné stránky budoucího produktu. Touto metodou je možné porovnávat svůj návrh s již existujícími konkurenčními produkty. Druhá metoda nazývaná zkráceně FMEA – Failure Mode and Effects Analysis znamená Analýzu příčin důsledků vad. Tato metoda analyzuje, jak je možné předcházet vadám, v našem případě by se jednalo o analýzu vhodnosti návrhu programu, tak aby byly ošetřeny všechny chyby, nebo program byl ještě doplněn o dodatečné funkce (vlastnosti). Obdobně jako první metoda je i tato založena na hodnocení potenciálních příčin, jejich výskytů, následků a četnosti odhalení. Obě tyto metody se tvoří týmově, a proto v této samostatné práci nebyly provedeny, ale přesto se staly podnětem při návrhu a zpracování tohoto projektu. [16]

Rozdělení úkolu do částí:

- 1) Návrh databáze
- 2) Návrh uživatelského rozhraní
- 3) Návrh vyhledávání v databázi
- 4) Návrh doplňujících funkcí programu
- 5) Tvorba lokalizačních balíčků

5.2 Návrh databáze

Databáze byla navržena s ohledem na požadavek zadání, kterým byla její co největší univerzálnost a dostupnost. Možnost přenositelnosti je důležitý z pohledu případného možného použití této databáze v budoucnu i pro jiné účely. Z možných databází jsem vybíral mezi MSSQL, MySQL, MS Access, případně se uvažovalo i o souborových formátech Exel nebo CSV případně i návrhu vlastního formátu. Vlastní formát jsem vyloučil jako první, protože nemá smysl tvořit něco co je již vytvořené. Souborové formáty Excel a CSV umí poměrně dobře zajistit přehlednou strukturu dat, ale případné vyhledávání a zobrazení není alespoň z mého pohledu optimální. Databáze MSSQL a MySQL jsou z pohledu strukturování dat, vyhledávání, prohlížení nebo propojení s jinými systémy téměř ideální, ale nesplňují podmínku jednoduché přenositelnosti. Obě dvě databáze standardně fungují na serverech, ale pokud by uživatel chtěl použít danou databázi na běžném (neserverovém) počítači, pak si musí nainstalovat MSSQL Server v případě databáze MSSQL nebo např. EasyPHP pro používání databáze MySQL. Tyto programy fungují spolehlivě, ale instalace a nastavení komunikace s databází může být zdlouhavé. Pro účely tohoto projektu jsem vybral databázi MS Access, přestože osobně nepovažuji MS Access za plnohodnotný nástroj v oblasti databází (na rozdíl od MSSQL nebo MySQL), pro naše účely je dostatečný a především plní požadavky. Dostupnost program MS Access je vysoká, protože je standardně prodáván jako součást balíku programů MS Office na platformě Windows. Jedná se o program umožňující tvorbu jednoduchých relačních databází, kde vlastní databáze je uložena v souboru s příponou mdb, což znamená Microsoft DataBase a tím, že se jedná o soubor, je zaručena jednoduchá přenositelnost mezi počítači s operačním systémem Windows. Provázání na jiné platformy je možné zajistit importováním tohoto souboru např. přes databázi MySQL. Takto vytvořená souborová databáze má i své nevýhody. Vzhledem k tomu, že se nejedná o klasickou databázi, není možné provádět do databáze paralelní zápis nebo při vyhledávání v databázi se vždy otevírá celý soubor, což zvyšuje zatížení serveru resp. procesoru. Tyto negativní vlastnosti umíme v našem případě obejít, první problém týkající se paralelního zápisu je významný při tvorbě webových aplikací, kdy do databáze má přístup několik uživatelů najednou, problém otevírání celého souboru se eliminovat nedá, ale vzhledem k tomu, že do budoucna pravděpodobně celkový počet položek nepřekročí 10.000 záznamů, nejedná se o velkou databázi, tedy hardwareové nároky nebudou vysoké.

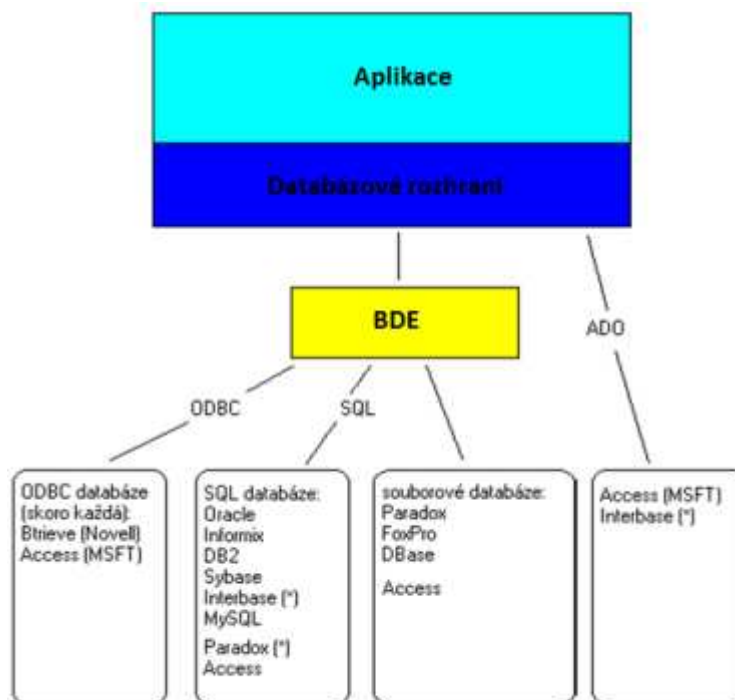
Vlastní databáze bude navržena pouze s jednou tabulkou bez dalších relačních vazeb. Na obrázku 4.2 je zobrazena struktura řádku ze souboru XML, odkud definujeme vlastní strukturu tabulky databáze. První sloupec tabulky bude obsahovat pořadové číslo záznamu, aby bylo možné jednoduše jednotlivé řádky identifikovat a proto byl nazván ID. Druhý sloupec bude obsahovat parametr "name" u tagu "item", sloupec byl nazván ITEM_NAME. Další dva sloupce budou obsahovat vlastní zprávu (originál) v angličtině a její překlad v češtině. Tyto sloupce byly nazvány TEXT_CZ a TEXT_EN. Na obrázku 5.1 je zobrazena tabulka s výše popsanou strukturou. Název tabulky je TAB_XML.

ID	ITEM_NAME	TEXT_CZ	TEXT_EN	KMM
1140	AboutWindowTitle	O Mediastite Recorder	About Mediastite Recorder	
948	AccessDeniedError	Přístup zamítnut.	Access is denied.	
75	ActiveServerConnectionDialogTitle	Aktivní připojení k serveru	Active Server Connection	
2234	AddAllowedConnection	Přidat povolené připojení	Add Allowed Connection	
2155	AddBroadcastAllowedConnection	Přidat povolené připojení	Add Allowed Connection	
2182	AddButtonText	Přidat	Add	
629	AddForumEmailText	Email:	Email:	
628	AddForumNameText	Jméno:	Name:	

Obrázek 5.1 Struktura databázového souboru

V okamžiku, kdy byl vytvořen typ a struktura databázové tabulky, zbývalo ji naplnit počátečními daty. Jako vstupní data byly použity soubory s příponou XML ze systému MediaSite. Ze všech souborů byly do tabulky zapsány položky do sloupce ITEM_NAME a TEXT_EN. Sloupec ID byl číslován automaticky (sloupec byl označen jako jedinečné automatické číslo). Prakticky byl tento přepis ze souborů do tabulky proveden automaticky pomocí programu, který byl k tomuto účelu vypracován. Tento program není podstatou této práce, a proto nebude dále nikde uveden. Vlastní překlad anglických textů byl proveden ručně pomocí internetového překladáče. [17]

Z hlediska databázového souboru je i důležité se zmínit o vlastním způsobu propojení s vlastním programem. Program je zpracován v C++ Builder, tento program umožňuje prostřednictvím komponent přistupovat i k databázovým objektům. Přístup k datům může být buď přímý, prostřednictvím technologie ADO, nebo přes rozhraní BDE. Struktura propojení je zobrazena na obrázku 5.2. Rozhraní BDE – Borland Database Engine umožňuje vytvořit



Obrázek 5.2 Databáze s aplikací

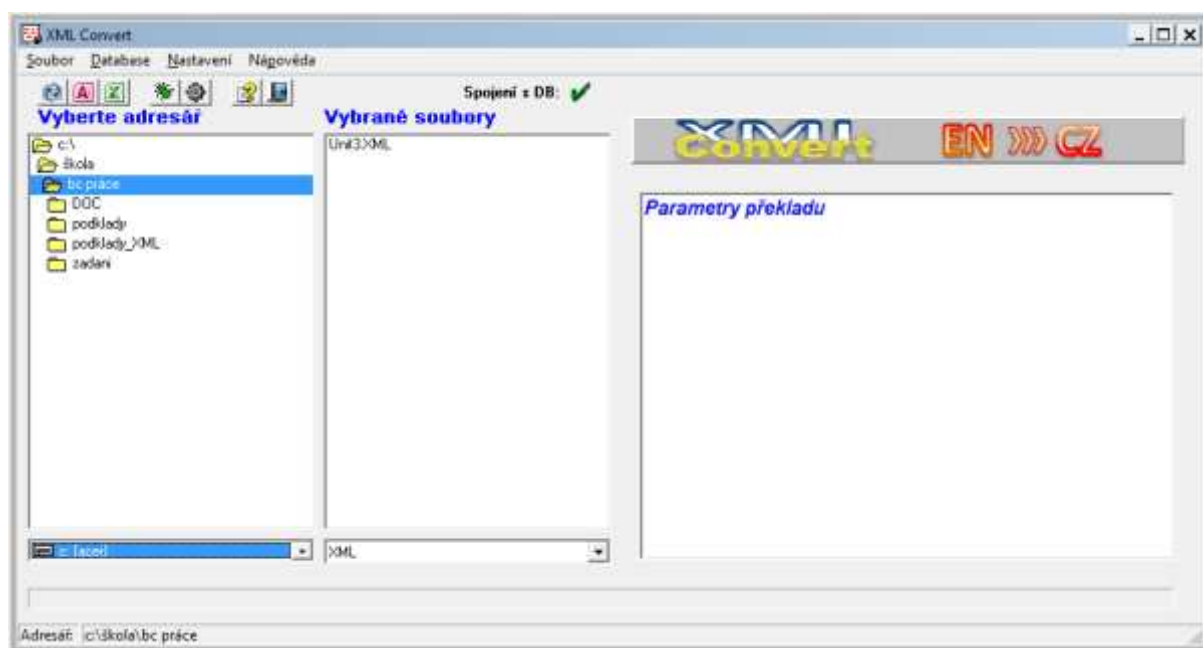
jedno univerzální rozhraní pro různé databáze, které mají rozdílné vlastnosti a tím umožnit programátorovi jednotný přístup ke všem databázím. Na první pohled ideální řešení má bohužel i své zápory. BDE je speciální program, který se na počítači chová jako server a během přístupu k databázi musí být spuštěn, což má za následek, že rozhraní se musí složitě nastavovat (je zde i možnost zjednodušení pomocí aliasů) a databázové soubory se musí instalovat. Program XML Convert využívá druhého způsobu spojení s databází. Technologie ADO – ActiveX Data Object, zajišťuje přístup k datovým zdrojům a vytváří rozhraní umožňující programátorovi přistupovat k databázím bez jejich hlubších znalostí. Programátor musí zajistit spojení s databází resp. tabulkou a pak už jen stačí přistupovat k jednotlivým záznamům. V tomto programu byly použity dvě komponenty, pro spojení s databází byla použita komponenta ADOConnection a pro komunikaci s daty komponenta ADOQuery, která umožňuje pracovat s databází prostřednictvím SQL příkazů.



Obrázek 5.3
ADO Komponent

5.3 Návrh uživatelského rozhraní

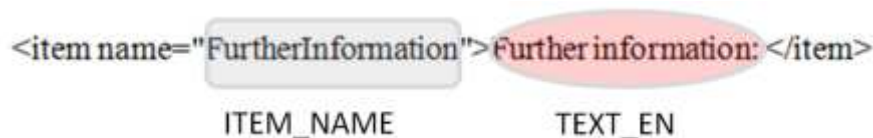
Návrh programu vychází z co nejjednoduššího a intuitivního ovládání. Program má zajistit převod souborů z angličtiny do češtiny, to znamená, že uživatel musí vybrat soubor a nechat ho přeložit. Vzhledem k tomu, že souborů k překladu bývá v adresáři většinou několik, stačí uživateli vybrat pouze adresář a program zpracuje všechny soubory v adresáři. Výsledné soubory jsou pak uloženy do stejného adresáře, jehož název je rozšířen o příponu _CZE. Na Obrázku 5.4 je zobrazeno základní okno programu.



Obrázek 5.4 Základní okno programu XML Convert

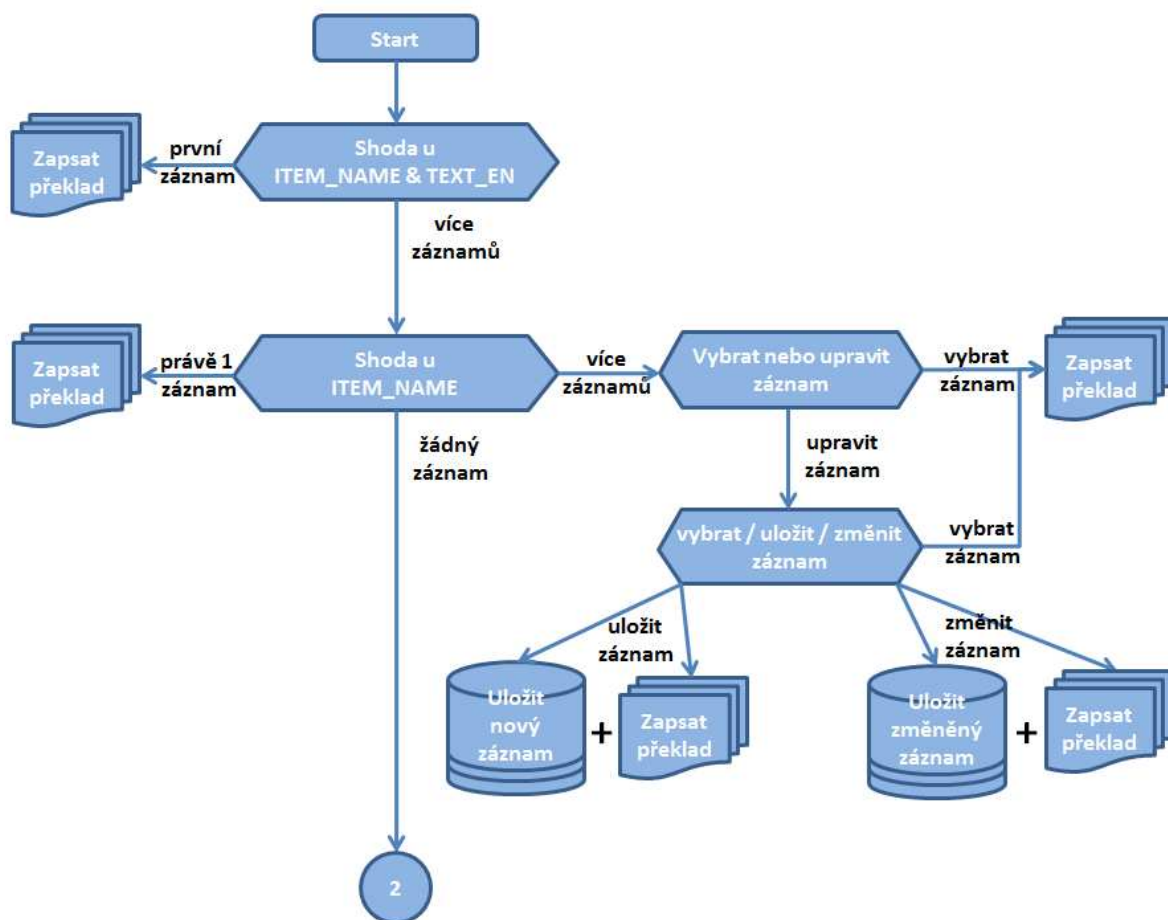
5.4 Návrh vyhledávání v databázi

Jednou z hlavních funkcí programu je algoritmus pro vyhledávání záznamů v databázi. Algoritmus musí být schopen nejen najít identické záznamy, ale i případné alternativy. Vyhledání identického záznamu je zřejmé, je dáno podmínkou shody parametru name v tagu item (zkráceně bude dále používáno pouze ITEM_NAME) a vlastního textu v anglickém jazyce (nadále bude zapisováno jako TEXT_EN) v překládaném souboru se stejnými proměnnými v databázovém souboru. Jednoznačný popis proměnných je na obrázku 5.5.



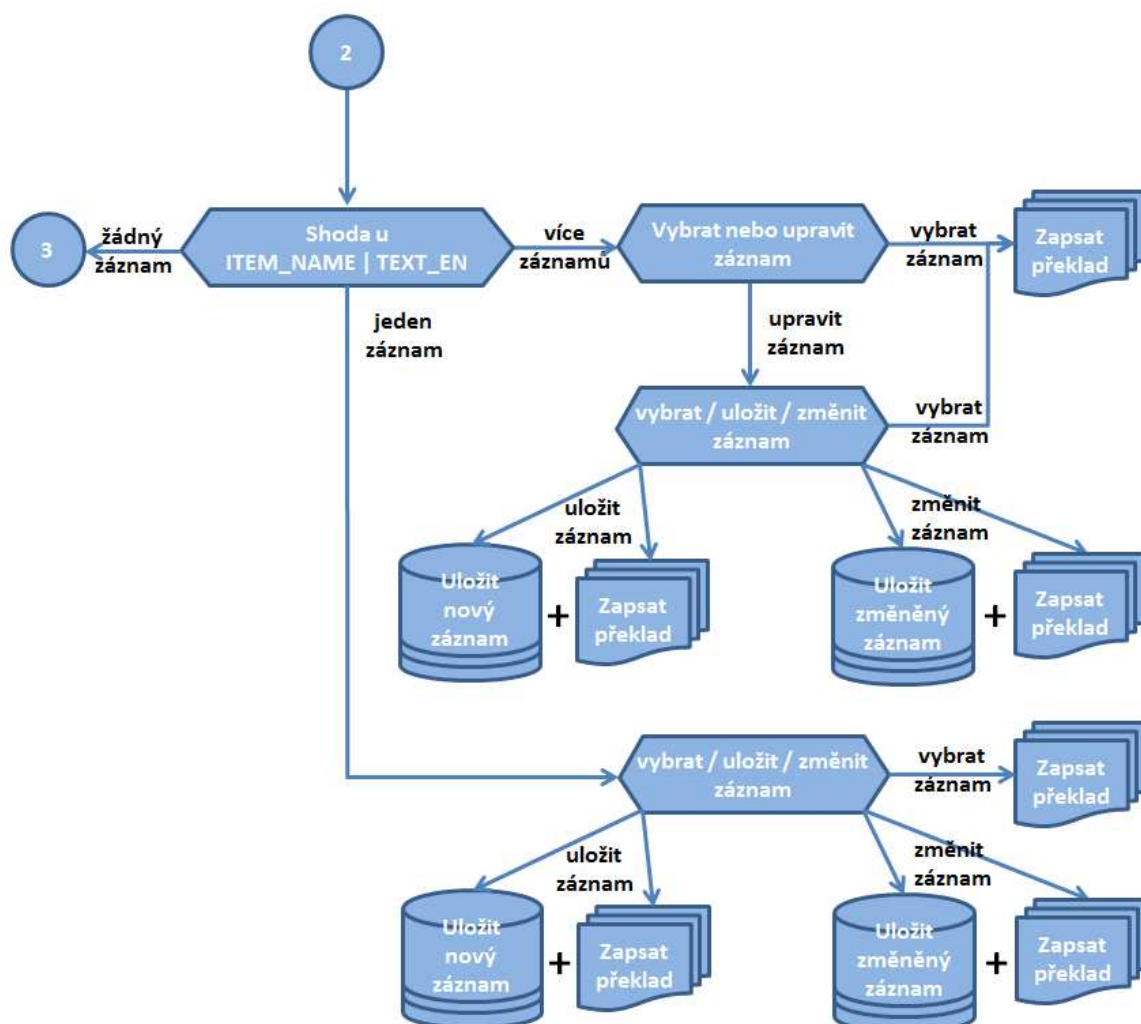
Obrázek 5.5 Vyznačení proměnných

Komplikovanější situace je v okamžiku, kdy výše uvedené shody není dosaženo. V tomto případě musí dojít ke kombinování možných variant, které mohou nastat. V případě, že není nalezena shoda, dojde k porovnání proměnných ITEM_NAME a pokud byl nalezen právě jeden záznam, pak je položka TEXT_EN nahrazena překladem TEXT_CZ. V případě, že bylo nalezeno více záznamů, pak se nabídne uživateli možnost výběru z těchto nalezených záznamů.



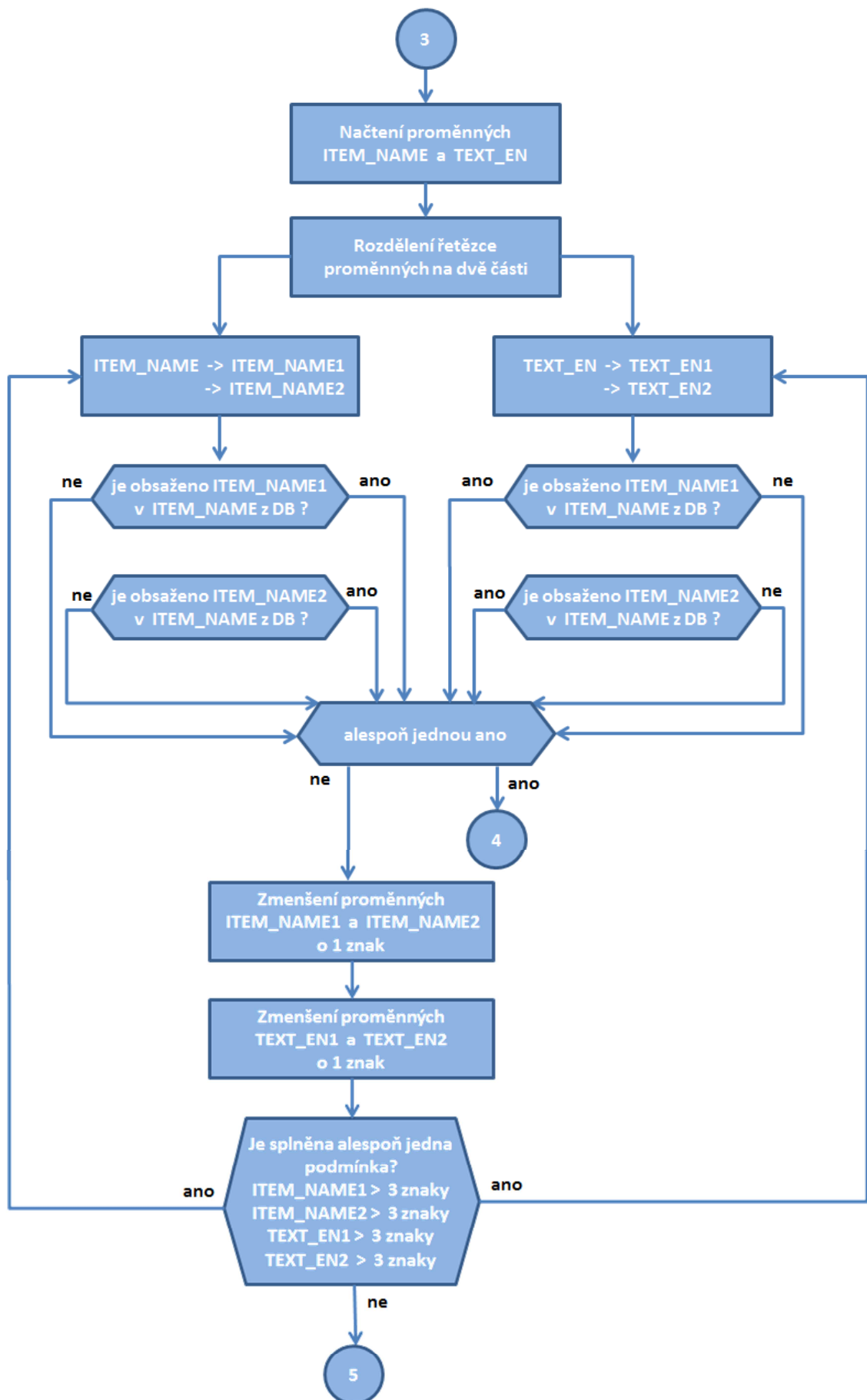
Obrázek 5.6a Vývojový diagram vyhledávacího algoritmu - část 1.

Naopak, to znamená, že nebyl nalezený žádný záznam, dojde k porovnání obsahu proměnných ITEM_NAME a TEXT_EN a stačí, aby byla splněna alespoň jedna rovnost, pak mohou nastat opět tři možnosti. Pokud byl nalezen jeden záznam, zobrazí se uživateli a je požadováno potvrzení správnosti překladu, v případě nalezení shody více souborů, opět se všechny možnosti nabídnou uživateli k výběru.



Obrázek 5.6b Vývojový diagram vyhledávacího algoritmu - část 2.

Jestliže nebyla nalezena shoda ani v tomto případě, bude program prohledávat podobnost proměnných ITEM_NAME a TEXT_EN tak, že vezme určité části řetězců a ty se budou srovnávat se záznamy v databázi. Prakticky to bude provedeno následovně, nejprve dojde k rozdělení proměnných překládaného souboru na polovinu, na dva stejně dlouhé řetězce (v případě lichého počtu znaků se budou řetězce lišit v délce o jeden znak) a tím nám přibudou dvě nové proměnné. Každou z nich budeme porovnávat se záznamy v databázi, samozřejmě bude porovnáváno, zdali záznam v databázi obsahuje tuto část řetězce. Jestliže shody nebylo nalezeno, pak se každý řetězec zmenší o jeden znak a celý proces hledání shody běží znova. Takový to cyklus probíhá, dokud není nalezen záznam odpovídající shodě, nebo dokud je délka řetězce větší než tři znaky, to platí pro všechny čtyři řetězce.



Obrázek 5.6c Vývojový diagram vyhledávacího algoritmu - část 3.

Prakticky si zmenšování proměnných o jeden znak ukážeme na výše uvedeném příkladu dle obrázku 5.5. Představme si, že v nějakém okamžiku bude proměnná ITEM_NAME obsahovat výraz “FurtherInfomation“ a proměnná TEXT_EN bude obsahovat “Furtherinfomation:“, pak po rozdělení každého z řetězců na dvě části získáme dvě nové proměnné (rozlišíme je indexem 1 resp. 2).

Výraz ITEM_NAME obsahující text “FurtherInfomation: “ se rozdělí na

ITEM_NAME1 = “FurtherI“ ITEM_NAME2 = “nfomation“.

Obdobně se rozdělí výraz TEXT_EN obsahující “FurtherInfomation: “ na

TEXT_EN1 = “FurtherIn“ TEXT_EN2 = “fomation:“.

Pokud nebyla ani teď nalezena shoda, pak dojde ke zmenšení všech řetězců o jeden znak a proměnné budou nově obsahovat následující hodnoty

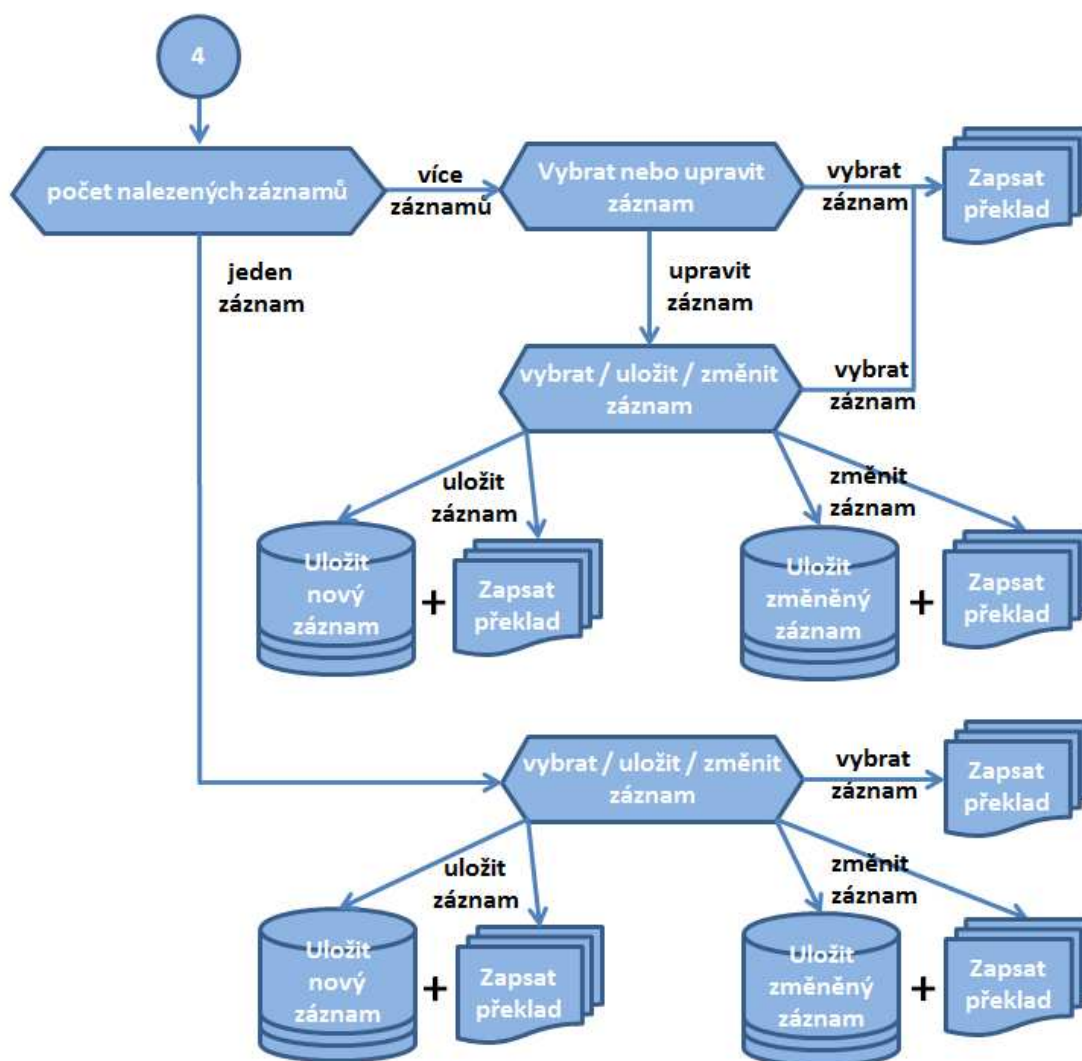
ITEM_NAME1 = “Further“ ITEM_NAME2 = “nfomatio“
TEXT_EN1 = “FurtherI“ TEXT_EN2 = “fomation “.

V případě, že se nedaří nalézt shodu se záznamem v databázi, pak celý cyklus kontroly pokračuje až do nejmenšího přípustného tvaru řetězců o délce tři znaky, proměnné pak obsahují

ITEM_NAME1 = “Fur“ ITEM_NAME2 = “nfo“
TEXT_EN1 = “Fur“ TEXT_EN2 = “fom“.

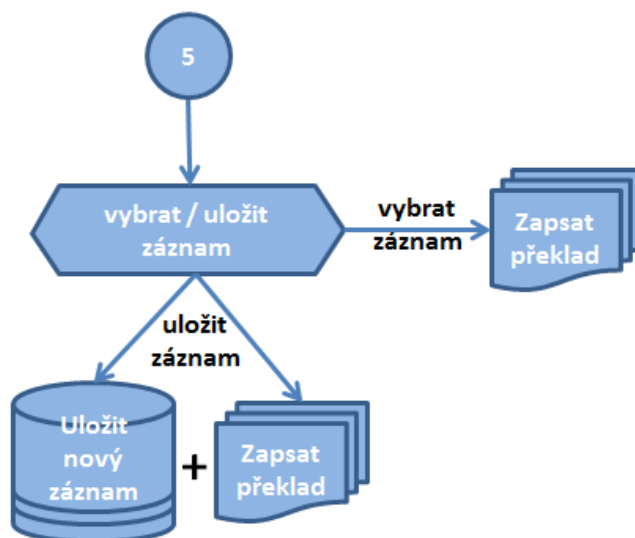
Pozorný čtenář si jistě všiml, že jednotlivé řetězce mají na počátku různou délku, přesto na konci je délka všech řetězců rovna třem. Toto je zajištěno adaptivním cyklem kontroly pro každý řetězec samostatně, aby bylo vyhodnocování jednotlivých řetězců na sobě nezávislé. Důvodem je možná situace, kdy např. výraz ITEM_NAME bude obsahovat 5 znaků a TEXT_EN bude obsahovat 80 znaků. Přestože výraz ITEM_NAME není možné po rozdělení již více zmenšovat (minimální počet znaků jsou tři), přesto celý cyklus a kontrola bude probíhat a zmenšování bude pouze řetězec TEXT_EN.

Je zřejmé, že čím méně znaků nám zůstává k hledání shody, tím horší bude kvalita nalezené shody. Proto při dosažení tří znaků již opětovná kontrola nepokračuje. Pokud nalezneme shodu v jednom nebo více záznamech, opět si uživatel může vybrat, resp. ověřit kvalitu navrhovaného překladu. Tento postup je zobrazen na obrázku 5.6d.



Obrázek 5.6d Vývojový diagram vyhledávacího algoritmu - část 4.

Pokud žádný ze záznamů v databázi neobsahuje žádnou z částí některého z řetězců, pak nebyla nalezena shoda výběru a uživatel je vyzván, aby doplnil požadovaný překlad. Jinými slovy, uživatel musí překlad provést ručně a je doporučeno takovýto nový záznam vždy uložit do databáze, aby při novém generování lokalizačního souboru nemusel uživatel opětovně tuto položku překládat.



Obrázek 5.6e Vývojový diagram vyhledávacího algoritmu - část 5.

Vlastní naprogramování je provedeno pomocí podmínek a cyklů v jazyce C++ v kombinaci s jazykem SQL. Vždy při provedení příkazu SQL se kontroluje počet nalezených příkazů příkazem `ADOQuery1->RecordCount`, kde první část představuje název třídy a druhá část název proměnné obsahující tuto informaci. Dále pokud uživatel vybírá některou z přednastavených možností počítačem, pak má vždy možnost příslušný záznam upravit, ale může i takto upravený záznam v databázi přepsat, nebo nechat vytvořit úplně nový záznam v databázi. Tím je umožněno rozšiřování databáze o nové záznamy, které se objevují v nových překládaných XML souborech. Prostředí programu, kde uživatel provádí výběr případně ukládání do databáze je zobrazeno na obrázku 5.7.

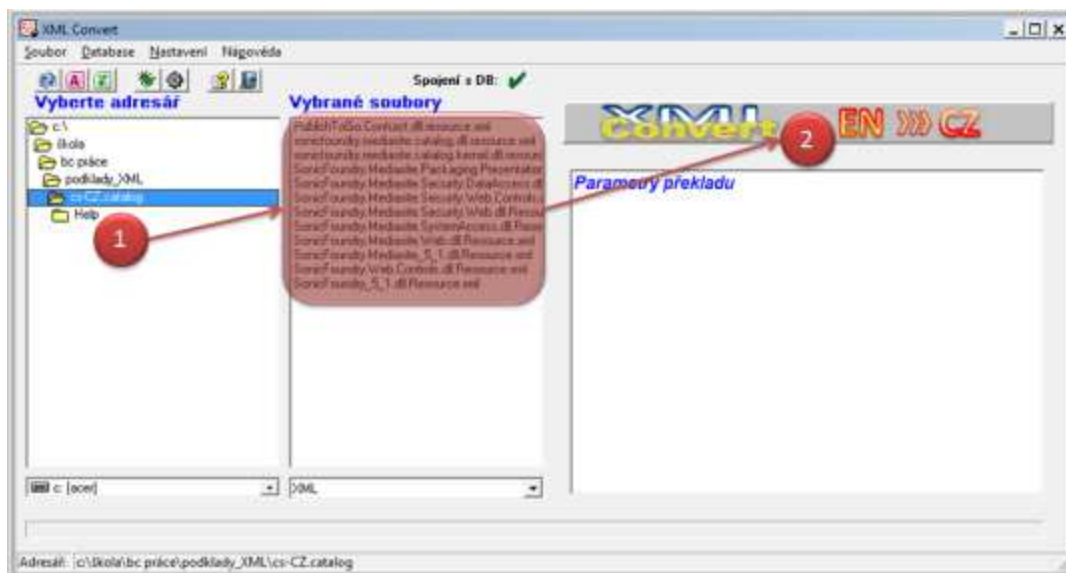
Bylo nalezeno více záznamů (2 záznamů)
vyberte vhodný záznam nebo ho upravte

Překládaný text <input type="text" value="XXX MenuItemControlsEncodeVideoInput"/> <input type="text" value="XXX Encode &amp;Video In"/>		<input checked="" type="checkbox"/> OK Odhadovaná shoda 50%	<input type="button" value="nový záznam"/> <input type="button" value="akt. záznam"/>
Návrh překladu <input type="text" value="MenuItemControlsEncodeVideoInput"/> <input type="text" value="Encode input &amp;video"/>		<input type="button" value="▲"/> <input type="button" value="▼"/>	<input type="text" value="Kodování vstupního &amp;video"/> Český název - překlad
<input type="text" value="<item name>"/>		Anglický název - originál	

Obrázek 5.7 Prostředí programu, kde probíhá překlad

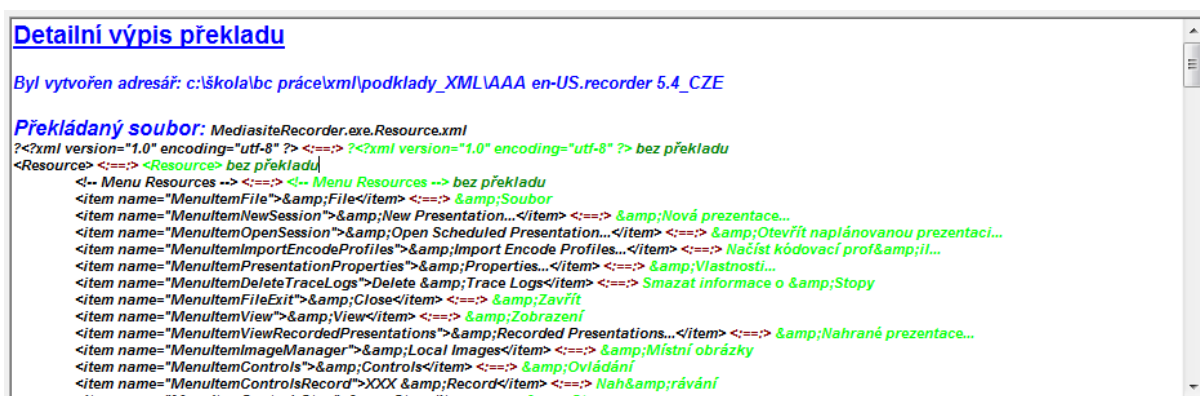
5.5 Popis programu XML Convert

Cílem návrhu programu byla snaha o jednoduché a intuitivní ovládání celého programu. Vlastní překlad se spouští ve dvou krocích. Nejprve uživatel vybere adresář, kde jsou soubory určené k překladu a následně zvolí vlastní překlad - viz obrázek 5.8.



Obrázek 5.8 Spuštění překladu

Detailní výpis překladu je možné sledovat ve spodním okně. Zde uživatel vidí všechny činnosti, které program vykonává. Nejprve je vytvořen adresář se stejným názvem jako je název vybraného adresáře uživatelem navíc rozšířeným o příponu “_CZE“. Do tohoto adresáře se ukládají přeložené soubory pod původním názvem, tedy stejným jako má originální soubor. Dále zde můžeme vidět jednotlivé řádky překládaného souboru a výsledek po překladu, zda byl přeložen případně vlastní překlad. Příklad výpisu překládaného souboru je vidět na obrázku 5.9.

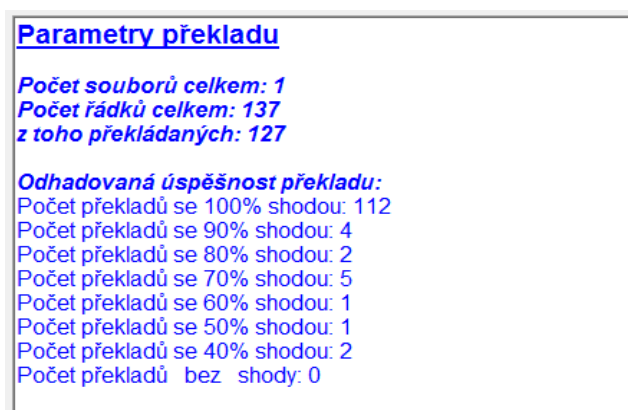


Obrázek 5.9 Výpis překládaného souboru

V případě, že není proveden automatický překlad, to znamená, že nebyla nalezena významná shoda mezi překládaným textem a textem již uloženým v databázi, má uživatel možnost si vybrat z předem navržených překladů, které mají nějakou podobnost

s překládaným výrazem. Algoritmus překladu a vyhledávání je popsán v kapitole 5.4, kde je i obrázek 5.7, který zobrazuje okno programu umožňující výběr z navržených překladů.

Na závěr překladu se vypíše report, který sumarizuje jednotlivé parametry překladu. Překlady se 100% a 90% pravděpodobností shody se provádí automaticky, u ostatních musí uživatel provést potvrzení navrženého překladu případně jeho korekci. Vlastní hodnota pravděpodobnosti shody byla stanovena empiricky a vychází z algoritmu popsaného v kapitole 5.4. Každá další výběr snižuje hodnotu pravděpodobnosti o 10%.



Obrázek 5.10 Závěrečná report překladu

Další informace o programu je možné nalézt v nápovědě, která je součástí programu.

6. Tvorba lokalizačních balíčků

6.1 Lokalizace prostředí

Pod pojmem balíček si můžeme jednoduše představit adresář s přeloženými soubory. Pokud bychom pouze tímto adresářem nahradili původní adresář v systému MediaSite, došlo by sice k lokalizaci portálu do Českého jazyka, ale ztratili bychom možnost spouštět program i v původním anglickém jazyce. Je výhodnější zachovat původní XML soubory v původním adresáři en-US a k němu vytvořit nový adresář cs-CZ, do kterého nakopírujeme přeložené XML soubory. Pokud pak bude systém MediaSite spouštět ve Windows s nastavenou výchozí češtinou, bude systém MediaSite zobrazovat texty v českém jazyce a naopak, pokud bude systém MediaSite spuštěn ve Windows s přednastavenou angličtinou, budou texty zobrazené v anglickém jazyce, což je hlavně přínosem pro zahraniční studenty.

Přeložené soubory byly uloženy do prostředí systému MediaSite, který běží na MediaSite serveru, který je základem portálu prednasky.tul.cz. V současnosti je na Technické univerzitě v Liberci instalována verze 5.5, ve které byly překlady odzkoušeny a ze které byly pořízené obrázky. Aktuálně nabízí firma Sonic Foundry (výrobce systému MediaSite) verzi MediaSite serveru 6.0.

Vzhledem k tomu, že jedním z cílů této práce je zvýšení uživatelského komfortu především pro běžného uživatele, byly přeloženy čtyři hlavní moduly, které uživatelé využívají. Každému modulu systému MediaSite odpovídá jeden z balíčků. Výsledné reporty překladů XML souborů jednotlivých balíčků, které byly vygenerované programem jsou v příloze č.2.

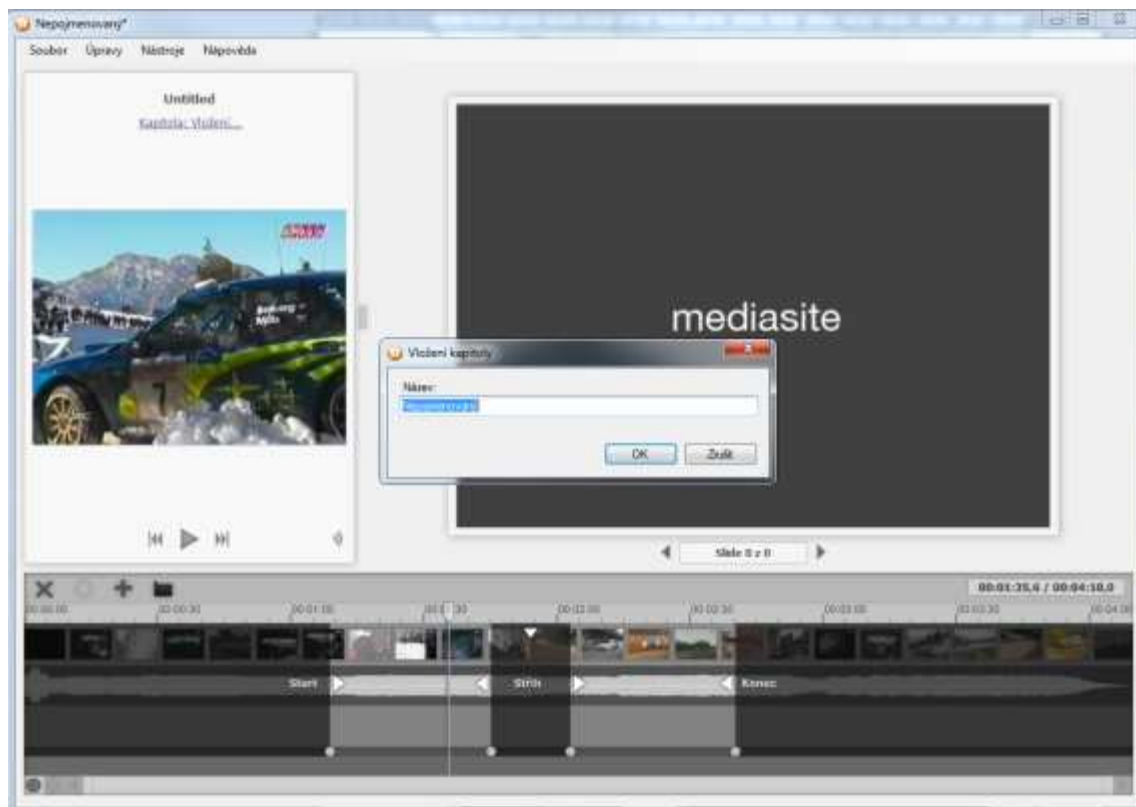
Názvy modulů a k nim překládané balíčky:

- | | |
|------------------|------------------------------|
| 1) Editor5.5 | - editace prezentací |
| 2) Catalog | - umožňuje práci s katalogem |
| 3) SSO LoginForm | - přihlášení do systému |
| 4) Viewer | - vlastní prohlížeč |

Následující části budou zobrazeny výsledky překladu, tak jak je uvidí uživatel při spuštění daného modulu systému MediaSite na portálu prednasky.tul.cz. Aby bylo možné výsledek porovnat, bude vždy uveden obrázek prostředí daného modulu v českém i anglickém jazyce. Jako první byl přeložen modul Editoru5.5, vlastní překlad proběhl bez problémů a výsledek je vidět na obrázcích 6.1 a 6.2. Česká lokalizace se projevuje v názvech, v menu a podmenu a v hlášeních.



Obrázek 6.1 Původní zobrazení v anglickém jazyce

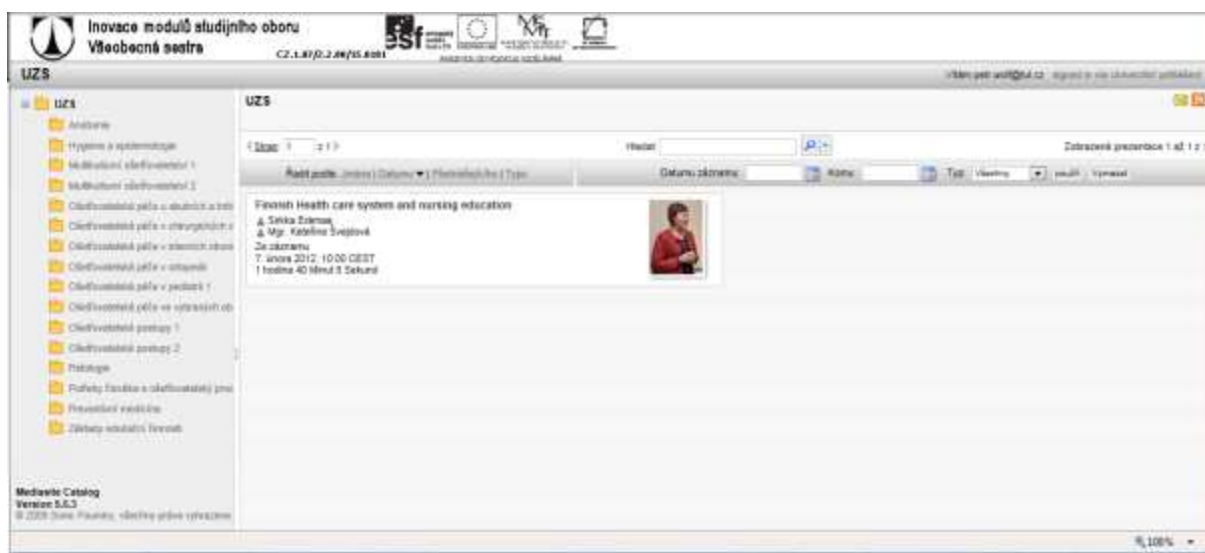


Obrázek 6.2 Nové zobrazení v českém jazyce

Dalším překládaným modulem, byl modul Catalog. Přeložením a implementováním XML souborů do modulu Catalog došlo k počestění těchto stránek. Jak se tyto změny projeví na přednáškovém portálu, je možné vidět na obrázku 6.3 a 6.4. Hlavní změny se projevují v popisu vlastního prostředí a informačních hlášeních.



Obrázek 6.3 Anglická lokalizace modulu Catalog



Obrázek 6.4 Česká lokalizace modulu Catalog

Následně byl překládán modul SSO Login. Při překládání tohoto modulu však došlo ke dvěma problémům. Na obrázcích 6.5 a 6.6 jsou zobrazeny obě jazykové mutace a navíc na

obrázku 6.6 jsou zakreslené již zmiňované problémy. Problémy s překladem spočívaly v tom, že po přeložení XML souborů zůstaly dva texty nepřeložené.

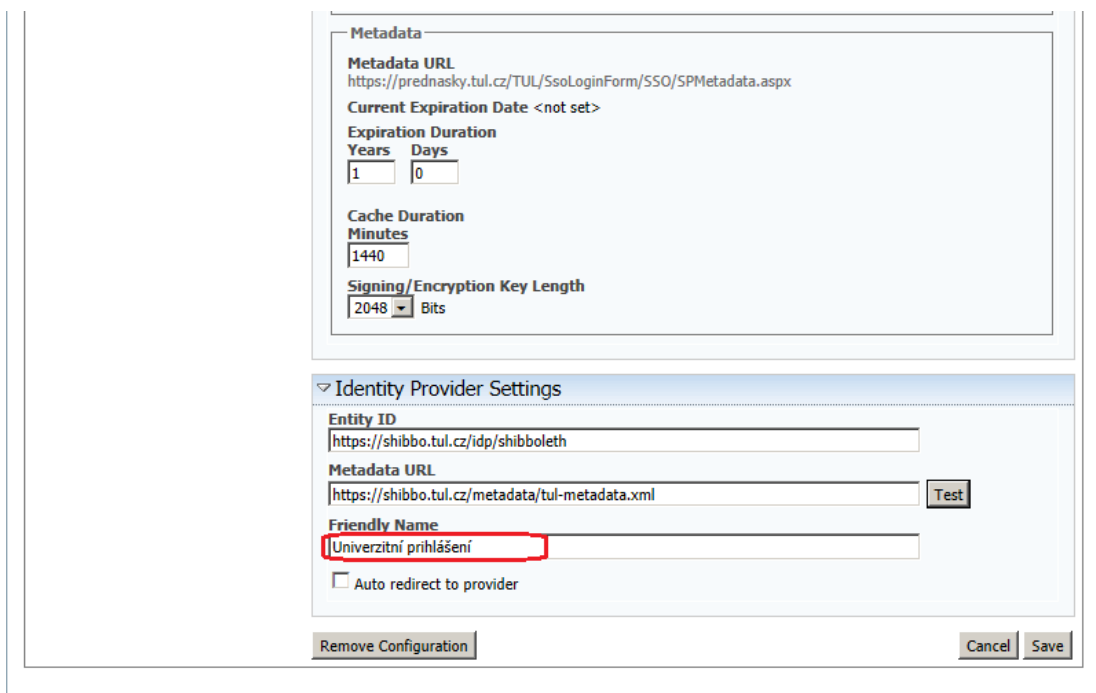


Obrázek 6.5 Anglická lokalizace modulu SSO Login



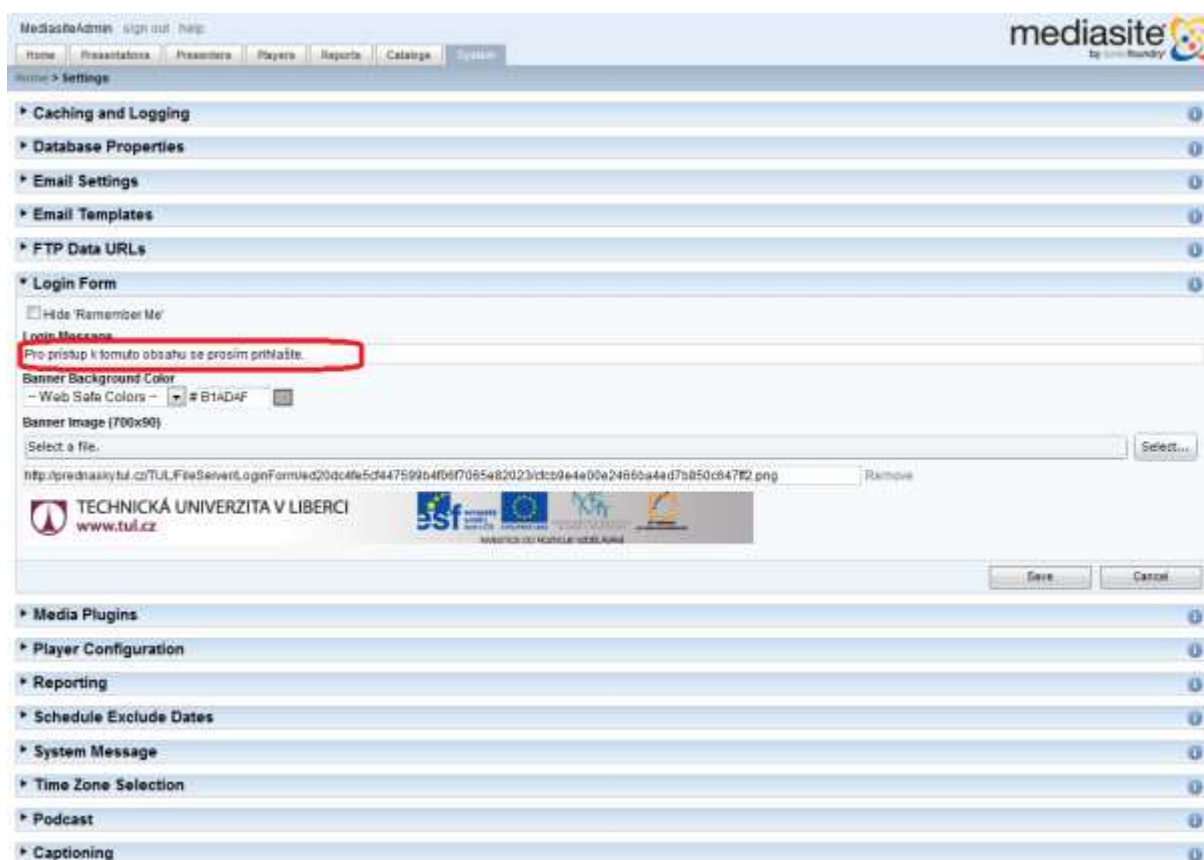
Obrázek 6.6 Česká lokalizace modulu SSO Login

Položky jako např. Uživatelské jméno nebo tlačítko Přihlásit se standardně překládají v XML souborech. Hlášení označená 1, 2 a 3 se ale překládají, alespoň z mého pohledu naprosto nepochopitelně, úplně jinde. Položka 1 - Univerzitní přihlášení se překládá v nastavení SSO Login v položce s názvem Friendly Name.



Obrázek 6.7 Česká lokalizace položky Univerzitní přihlášení

Obdobně bylo potřeba najít, kde se v systému MediaSite překládá druhá výše zmíněná položka. Navíc si pozorný čtenář jistě všiml, že tato položka je sice přeložena do českého jazyka, ale bez diakritiky. Celý tento problém byl komunikován přímo s výrobcem, s firmou Sonic Faundry, konkrétně se systémovým inženýrem Tedem Hochem (číslo dotazu case:00023784). Výsledkem celé komunikace bylo jednak nalezení umístění položky, tato položka se mění v nastavení systému pod položkou s názvem Login Message – viz obrázek 6.8 a zároveň výrobce uvedl, že verze MediaSite serveru 5.5, která je v současnosti instalována na Technické univerzitě v Liberci a využívána portálem prednasky.tul.cz, neumožňuje podporu češtiny. Dle vyjádření výrobce je podpora diakritiky možná až od verze MediaSite serveru 6.0, které je již na trhu.



Obrázek 6.8 Česká lokalizace položky Univerzitní přihlášení

Položka 3 - Ověřit přes: se překládá v souboru Login.aspx. V tomto modulu se již jedná o čtvrté místo, kde jsou texty uloženy. Struktura a vyznačení hledaného textu v tomto souboru je na obrázku 6.9.

```

1 <% Page Language="C#" AutoEventWireup="true" CodeBehind="Login.aspx.cs" Inherits="SonicFoundry.Mediasite.Security.SsoLoginForm.Login" Theme="Standard" %>
2
3 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml" >
6 <head runat="server">
7   <title>Mediasite Login</title>
8 </head>
9 <body>
10   <form id="LoginForm" method="post" runat="server">
11     <div>
12       <div id="dvTopBanner" class="topBannerDiv" runat="server">
13         <asp:Image ID="BannerImage" SkinID="LoginBanner" AlternateText="" runat="server" />
14       </div>
15       <div class="loginContainer">
16         <div id="loginFormDiv" class="loginFormContainer">
17           <div id="loginControlContainer" runat="server" class="loginControlContainer">
18             <asp:Login ID="LoginControl" runat="server" OnLoggedIn="LoginControl_LoggedIn" OnLoginError="LoginControl_LoginError"/>
19           </div>
20           <div id="ssoRedirectContainer" runat="server" class="ssoRedirectContainer">
21             <asp:Label ID="SsoTitleLabel" CssClass="ssoTitle" runat="server" Text="Ověřte se, prosím:" />
22             <asp:Label ID="IdpFriendlyName" CssClass="ssoFriendlyName" runat="server" />
23             <div class="redirectButtonContainer">
24               <asp:Button ID="RedirectButton" runat="server" OnClick="RedirectButton_Click" />
25             </div>
26           </div>
27         </div>
28       </div>
29       <asp:Panel ID="NotificationPanel" runat="server">
30         <div class="notificationBox" runat="server" id="notificationBox">
31           <asp:Label ID="labelNotify" runat="server" CssClass="feedback" />
32         </div>
33       </asp:Panel>
34     </div>
35   </form>
36 </body>
37 </html>

```

Obrázek 6.9 Hledaný text v souboru Login.aspx

Posledním modulem k překladu je modul Viewer. I při překladu tohoto modulu se objevilo neočekávané překvapení. Některé položky se opět nepřekládají standardně v XML souborech, ani se nepřekládají v nastaveních, ale jsou uloženy v souboru Localization.js. Opět si na tomto místě neodpustím poznámku o nepochopitelné nekoncepčnosti při návrhu struktury hlášení v systému MediaSite, neboť v každém modulu mít hlášení uložená na různých místech nebo v různém formátu je pro programátory zbytečně nepřehledné a pro správce systému MediaSite přinejmenším obtěžující.

```

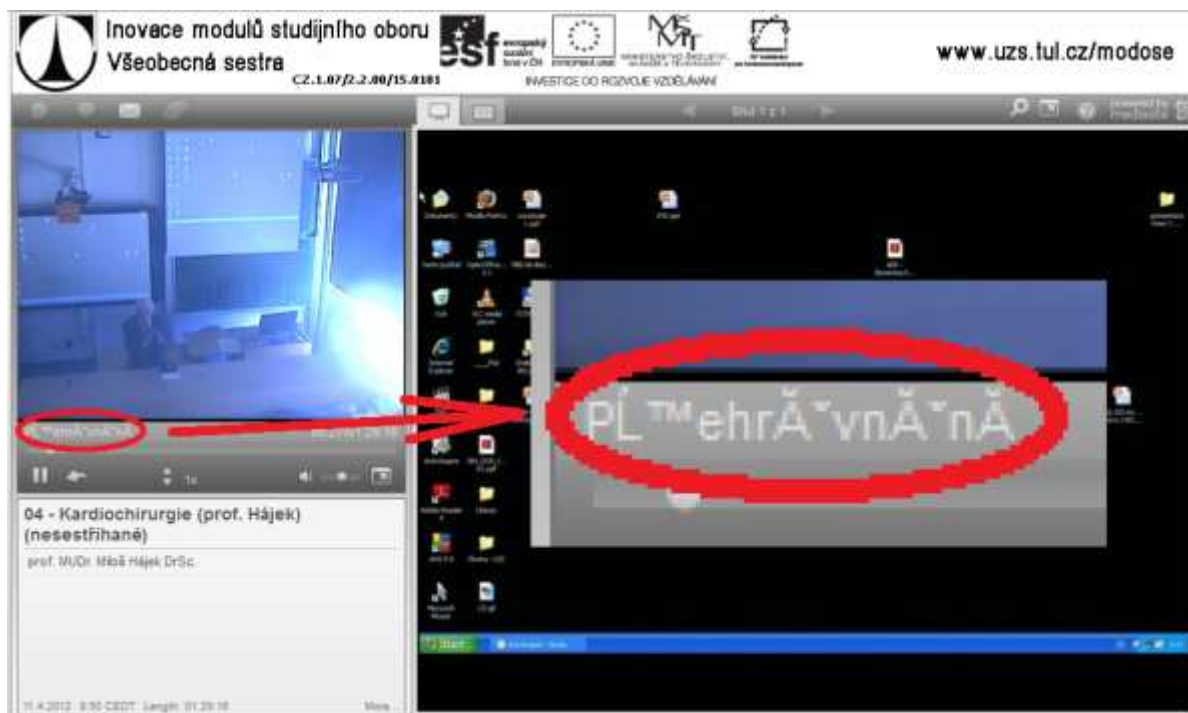
1 Localization.ThumbnailsResource=function(){}
2 Localization.LiveIndicatorResource=function(){}
3 Localization.PlayerLayoutResource=function(){}
4 Localization.PresentationCardResource=function(){}
5
6 Localization.Common.registerClass('Localization.Common');
7 Localization.Buttons.registerClass('Localization.Buttons');
8 Localization.FullSize.registerClass('Localization.FullSize');
9 Localization.CurrentSlideResource.registerClass('Localization.CurrentSlideResource');
10 Localization.LinksResource.registerClass('Localization.LinksResource');
11 Localization.ThumbnailsResource.registerClass('Localization.ThumbnailsResource');
12 Localization.PlayerLayoutResource.registerClass('Localization.PlayerLayoutResource');
13 Localization.PresentationCardResource.registerClass('Localization.PresentationCardResource');
14 Localization.LiveIndicatorResource.registerClass('Localization.LiveIndicatorResource');
15
16 Localization.Common.Slide = 'Slide';
17 Localization.Common.Of = 'of';
18 Localization.Common.PreviousSlide = 'Previous Slide';
19 Localization.Common.NextSlide = 'Next Slide';
20 Localization.Common.Language = 'Language';
21 Localization.Common.AudioTrack = 'Audio Track';
22 Localization.Common.Expand = 'Expand';
23 Localization.Common.Collapse = 'Collapse';
24 Localization.Common.SlideShow = 'Slide Show';
25 Localization.Common.SlideList = 'Slide List';
26 Localization.Buttons.ShowPolls = 'Show Polls';
27 Localization.Buttons.Ask = 'Ask a Question';
28 Localization.Buttons.Email = 'Share Presentation';

```

Obrázek 6.10 Struktura souboru Localization.js

Na obrázku 6.10 je zobrazena část struktury souboru Localization.js. Jedná se o soubor JavaScriptu a jsou v něm uloženy jednak položky podporující vlastní program (na obrázku 6.10 jsou vyznačeny červeně) a ty se samozřejmě nepřekládají a zbývající položky obsahují hledané texty a ty je potřeba přeložit.

Pokud jednoduše přeložíme položky do češtiny, narazíme na další problém. Program nepodporuje češtinu – viz obrázek 6.11.



Obrázek 6.11 Problém při překladu souboru Localization.js

Vzhledem k tomu, že jsme vlastními silami nedokázali vyřešit tento problém, tak jsme se opět obrátili na technickou podporu firmy Sonic Foundry. Pod číslem dotazu case: 00023785 jsme dostali informaci od pana Teda Hocha, že musíme nakopírovat tento soubor do příslušného adresáře.

```
..\Viewer\Localization\en-US\Localization.js  
..\Viewer\Localization\cs-CZ\Localization.js
```

Po odzkoušení se ovšem ukázalo, že jen toto nestačí a po další komunikaci s firmou Sonic Foundry bylo doporučeno ukládat soubory jako “Unicode“. Výsledek překladu modulu Viewer je na obrázcích 6.12 a 6.13.



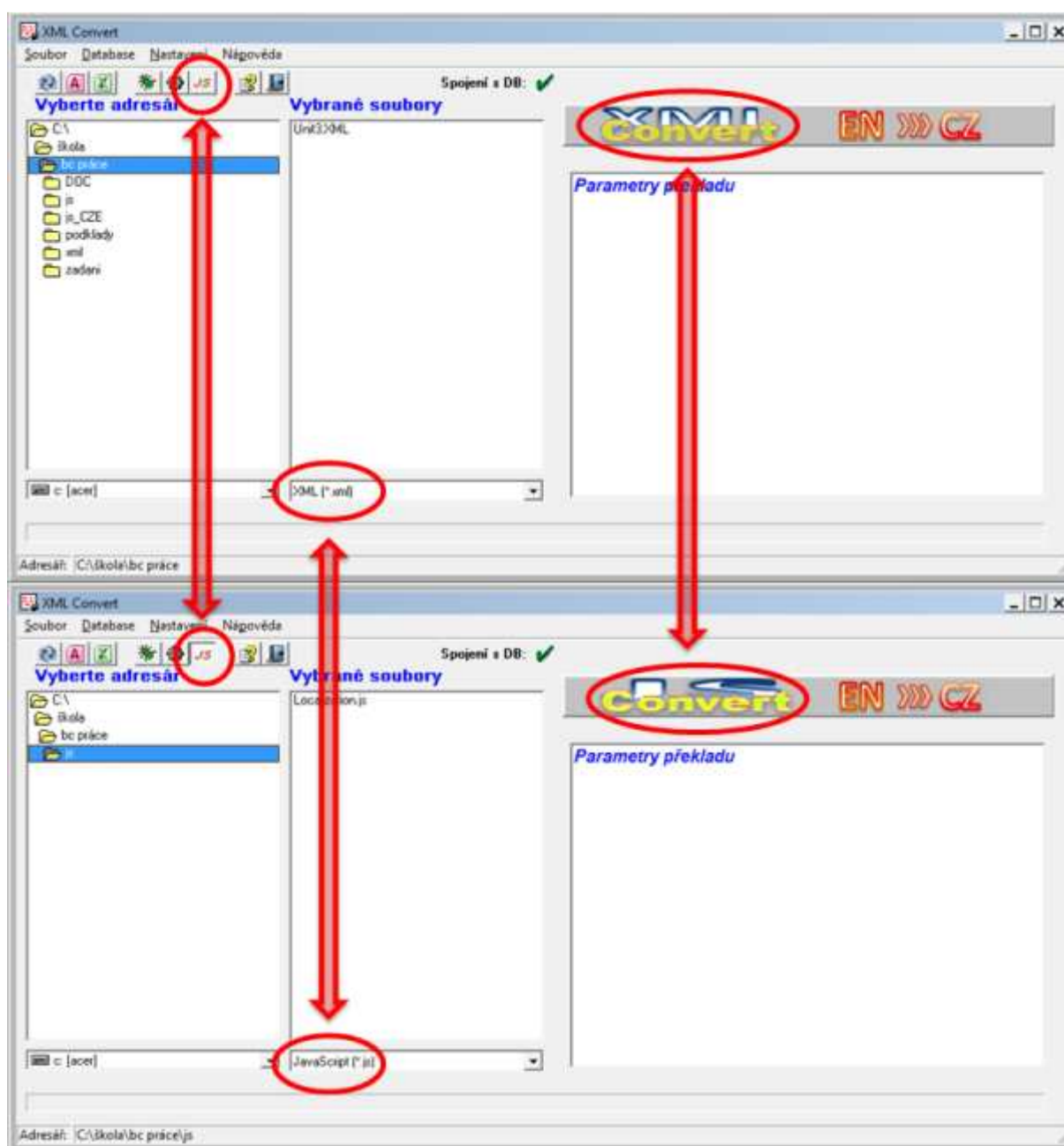
Obrázek 6.12 Anglická lokalizace modulu Viewer



Obrázek 6.13 Česká lokalizace modulu Viewer

Jako reakce na problém s překladem JavaScriptového souboru byl upraven i program XML Conver tak, aby byl schopen konvertovat i tyto soubory. V horní liště přibyl tlačítko s označením JS, které přepíná režim překladu mezi formátem XML a JavaScriptem. Jak se změny projeví v programu je vidět na obrázku 6.14.

6.2 Rozšíření programu o překlad JavaScriptových souborů



Obrázek 6.14 Změny v programu po rozšíření o překlad JavaScriptu

Vnější prostředí programu doznalo minimálních úprav, které jsou patrné z obrázku 6.14. Vlastní překlad JavaScriptových souborů, tedy algoritmus vyhledávání v databázi, nastavení, výběr z doporučených překladů nebo výstupní reporty vypadají a probíhají úplně stejně jako u překladu XML souborů, který byl detailně popsán výše. To samé platí i o datové struktuře uložené v databázi. Pouze pod položkou ITEM_NAME (viz obrázek 5.1) se v databázi ukládá název JavaScriptového objektu, např. Localization.Common.Slide (viz obrázek 6.10). Výsledek překladu souboru Localization.js je zobrazen v příloze č.3.

7. Závěr

Tato práce se zabývá lokalizací přednáškového portálu `prednasky.tul.cz` do českého jazyka a cílem práce je seznámit čtenáře s možnostmi lokalizace portálů, které využívají vícedruhových médií systému MediaSite, tak jako Technická univerzita v Liberci. Práce popisuje způsob, jakým systém MediaSite zaznamenává svá hlášení, která jsou předmětem jazykové lokalizace. Aby bylo možné spustit automatizovaný překlad souborů, byla nejprve vytvořena překladová databáze. Tato databáze obsahuje názvy hlášení, originální texty a jejich překlady. Následně byl vytvořen program XML Convert, který prostřednictvím překladové databáze překládá originální hlášení do Českého jazyka. Jako součást programu byl navržen vyhledávací algoritmus, který umožňuje jednoduchou formou provádět automatizovaný překlad tak, aby byl pro uživatele co nejpohodlnější a nejrychlejší. Přeložením jednotlivých souborů a nahrazením v systému MediaSite došlo k vlastní lokalizaci daného portálu. Vlastní překlad však neproběhl bez problémů. U některých modulů se objevily položky, které po překladu XML souborů nebyly přeloženy. Tento problém jsme nedokázali vyřešit vlastními silami, a proto jsme tento problém řešili přímo s výrobcem, s firmou Sonic Foundry. Trochu nepochopitelně se některé položky musí překládat v nastavení modulu, nebo v nastavení systému MediaSite a jiné v JavaScriptovém souboru. Položky měněné v nastavení systému MediaSite se nedají měnit automaticky, jedná se celkem jen o tři položky. Pro překlad JavaScriptových souborů byl dodatečně program XML Convert rozšířen i možností překladu JavaScriptových souborů. Vlastní překlad probíhá úplně stejně jako překlad XML souborů, ve stejném prostředí, využívá stejného vyhledávacího algoritmu a i datové položky jsou uloženy ve stejné databázi.

Úplnost naplnění cílů této práce ukáže až čas, ale už teď je zřejmé, že vlastní lokalizace proběhla úspěšně a vytvořený program s překladovou databází umožňuje generování lokalizačních balíčků při jakékoli aktualizaci systému MediaSite a díky tomu je možné usnadnit práci s portálem běžným uživatelům, učitelům nebo studentům, případně pomáhá také studentům na univerzitě třetího věku nebo handicapovaným studentům.

Následným krokem této práce by mohlo být implementování databáze i pro jiné portály nebo produkty, které nevyužívají datové struktury pro uložení svých hlášení ve formátu XML resp. rozšíření programu XML Convert o výstupní modul, který by umožňoval načítání a export překládaného souboru i do jiných formátů než jen XML. Další modifikací by bylo možné upravit program na internetové rozhraní tak, aby byl volně dostupný a využitelný i pro ostatní uživatele systému MediaSite.

Seznam literatury

- [1] <http://www.sonicfoundry.com/MediaSite>
- [2] <http://www.art-vision.eu/media/files/sofo-1.10-recorderczts.pdf>
- [3] XML pro každého, podrobný průvodce, Jiří Košek, Grada 2000
- [4] XML krok za krokem, Michael J. Young, idnes internet knihy
- [5] <http://www.w3.org/>
- [6] Wiener, Norbert. Kybernetika a společnost. Praha: Academia, 1963. 1. vyd. 216 s
- [7] Slámová, Hana. Informace a společnost, dostupné na World Wide Web:
<http://www.joomla.slamow.com/informace-a-spolecnost/prednasky/83-informace-a-spolecnost-i-uvod.html>
- [8] http://charita.xf.cz/tfjcu/skripta_TAMSP.pdf
- [9] <http://slovník-cizich-slov.abz.cz/web.php/slovo/mehrabian-albert>
- [10] http://cs.utmerketartikkel.com/Face-to-Face-sdělení-5-důvodů-proč-je-stále-důležité_823322
- [11] http://is.muni.cz/th/180349/ff_m_a2/dp.txt
- [12] <http://is2.lse.ac.uk/asp/aspecis/20040155.pdf>
- [13] <http://blog.timebarrow.com/2009/09/media-richness-theory/>
- [14] http://en.wikipedia.org/wiki/Media_richness_theory
- [15] http://en.wikipedia.org/wiki/Media_naturalness_theory
- [16] <http://www.chaloupka-kvalita.cz/fmea>
- [17] <http://translate.google.com/#>
- [18] <http://www.fsid.cvut.cz/cz/u12110/Pgs/PREDNESY/02/index.htm>
- [19] <http://www.cs.vsb.cz/behalek/vyuka/pcsharp/text/ch07s01.html>
- [20] <http://www.washington.edu/accessit/articles?1146>
- [21] <http://www.amx.com/>

Seznam příloh

Příloha č.1 Seznam přiložených souborů

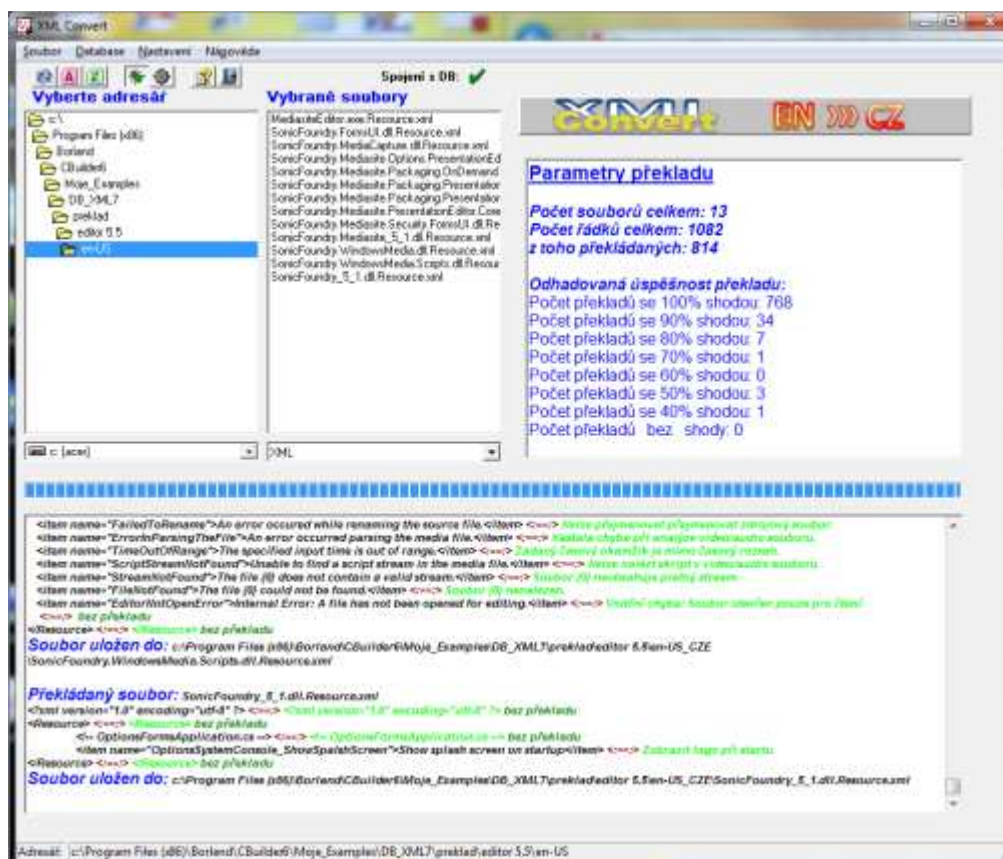
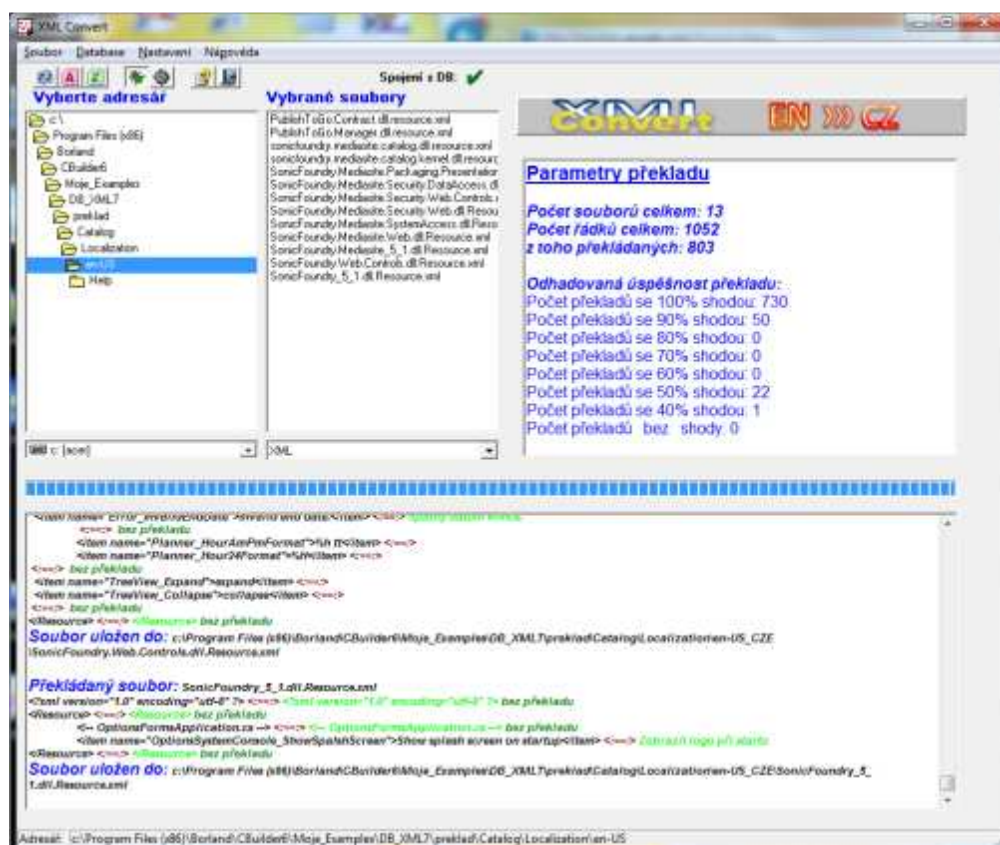
Příloha č.2 Výsledky překladu lokalizačních balíčků - XML soubory

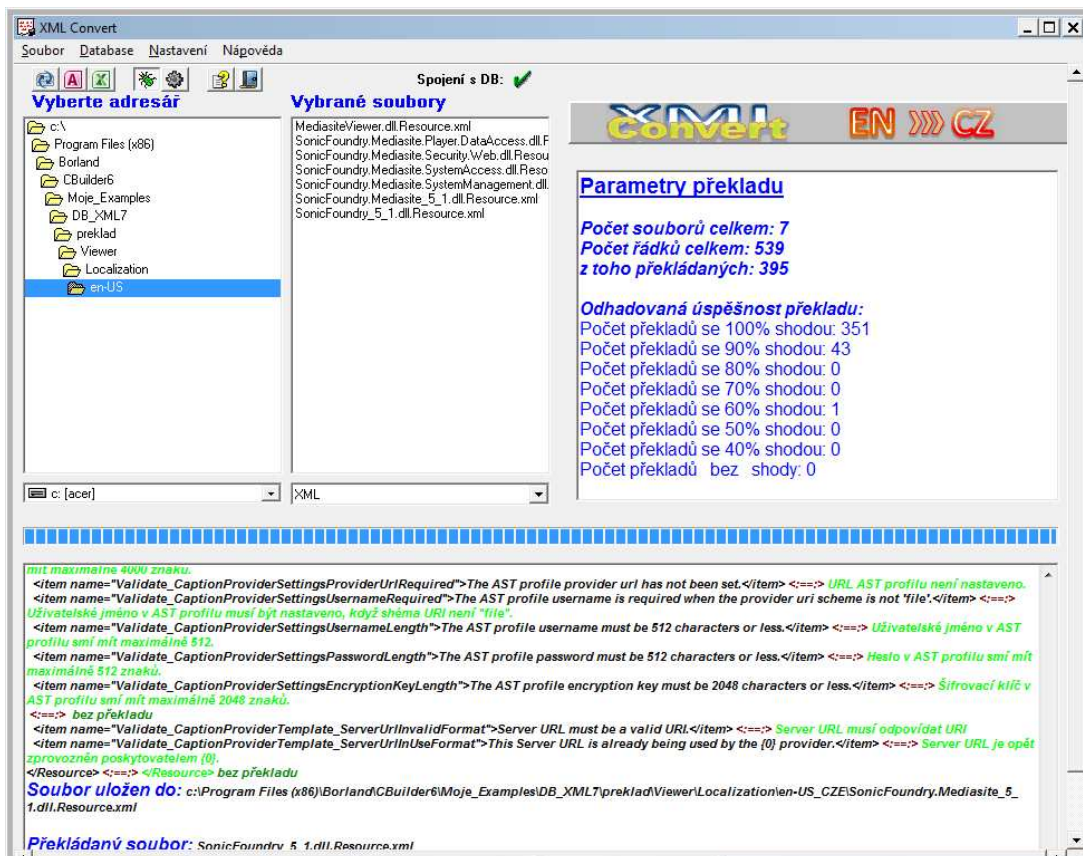
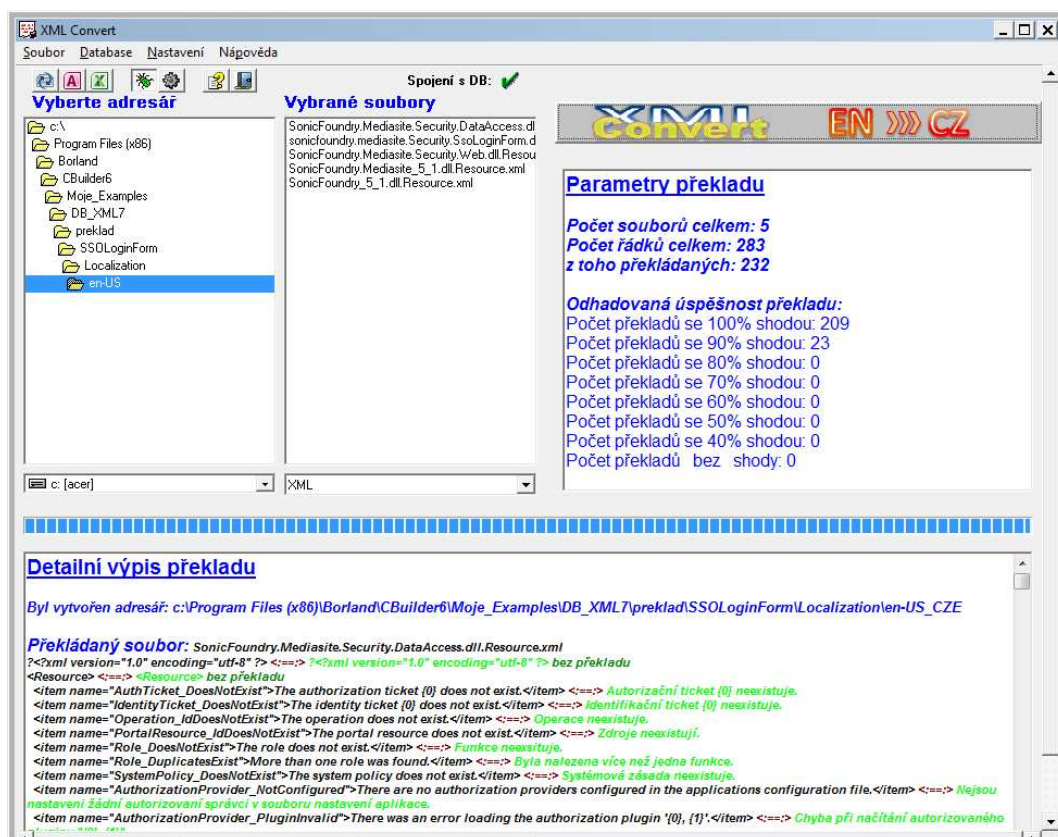
Příloha č.3 Výsledky překladu souboru Lokalization.js

Příloha č.1 Seznam příložených souborů – XML soubory

Adresář:	JS cs-CZ	obsahuje přeložený JavaScriptový soubor
Adresář:	XML Convert – Program	obsahuje instalační soubor programu
Adresář:	XML Convert – Source	obsahuje zdrojové soubory programu
Adresář:	XML cs-CZ	obsahuje přeložené XML soubory
Adresář:	XML en-US	obsahuje původní XML soubory

Příloha č.2 Výsledky překladu lokalizačních balíčků





Příloha č.3 Výsledky překladu souboru Lokalizace.js

